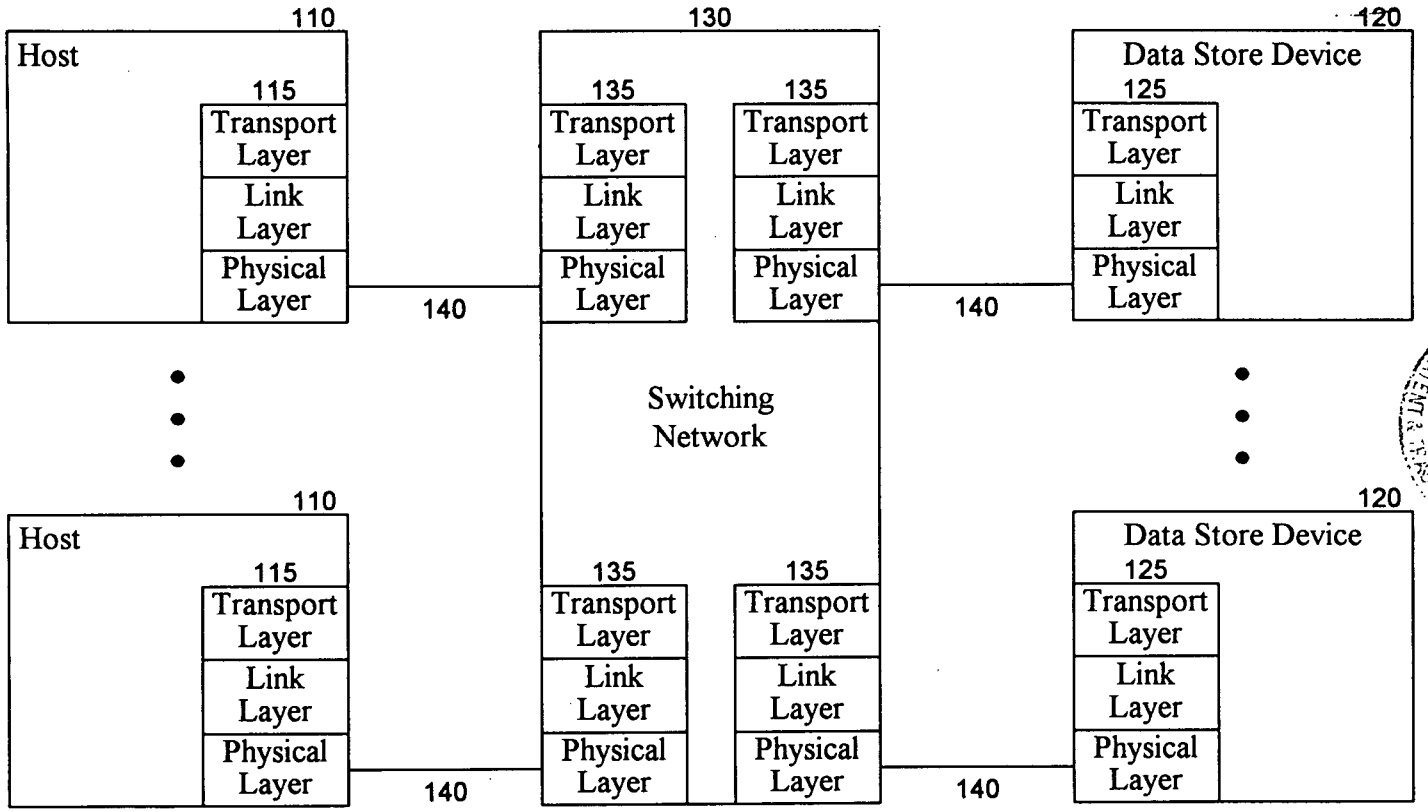


Fig. 1

PATENT & TRADEMARK OFFICE
MAY 10 2002
OIPE
JC73

COPY OF PAPERS ORIGINALLY FILED

10052464 - 051002

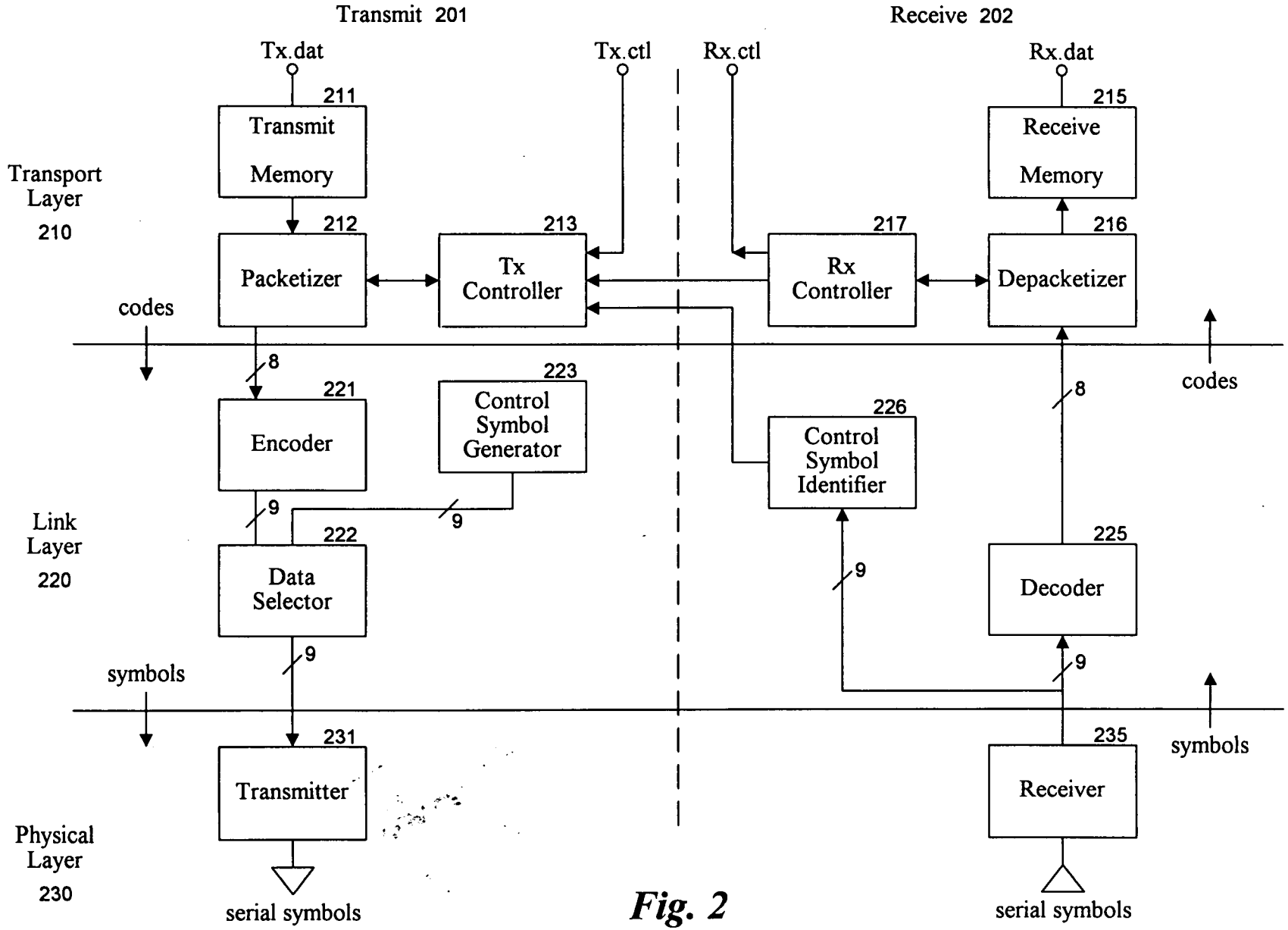


Fig. 2

EXHIBIT 1005 - 0151002

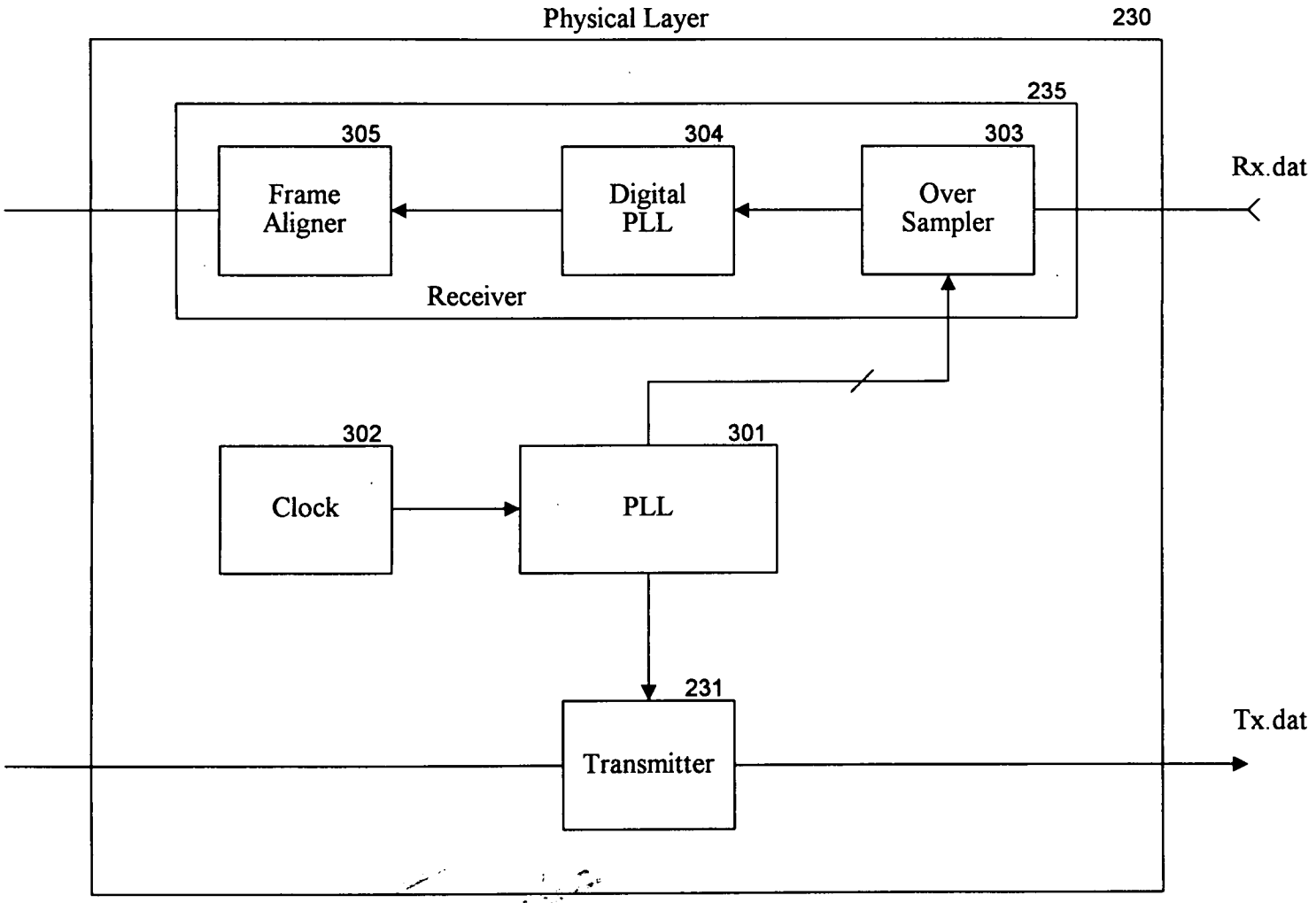

Physical Layer
230
235
305
Frame Aligner
304
Digital PLL
303
Over Sampler
Rx.dat
Receiver
302
Clock
301
PLL
231
Transmitter
Tx.dat

Fig. 3

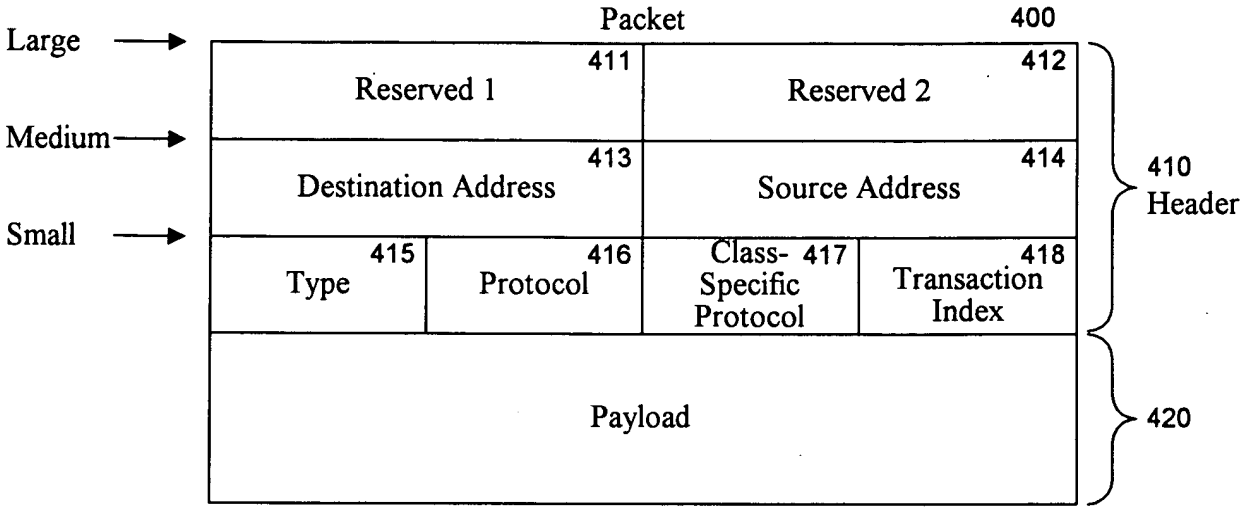

Large
Medium
Small
Packet
400
411
Reserved 1
412
Reserved 2
413
Destination Address
414
Source Address
415
Type
416
Protocol
417
Class-Specific Protocol
418
Transaction Index
410
Header
Payload
420

Fig. 4

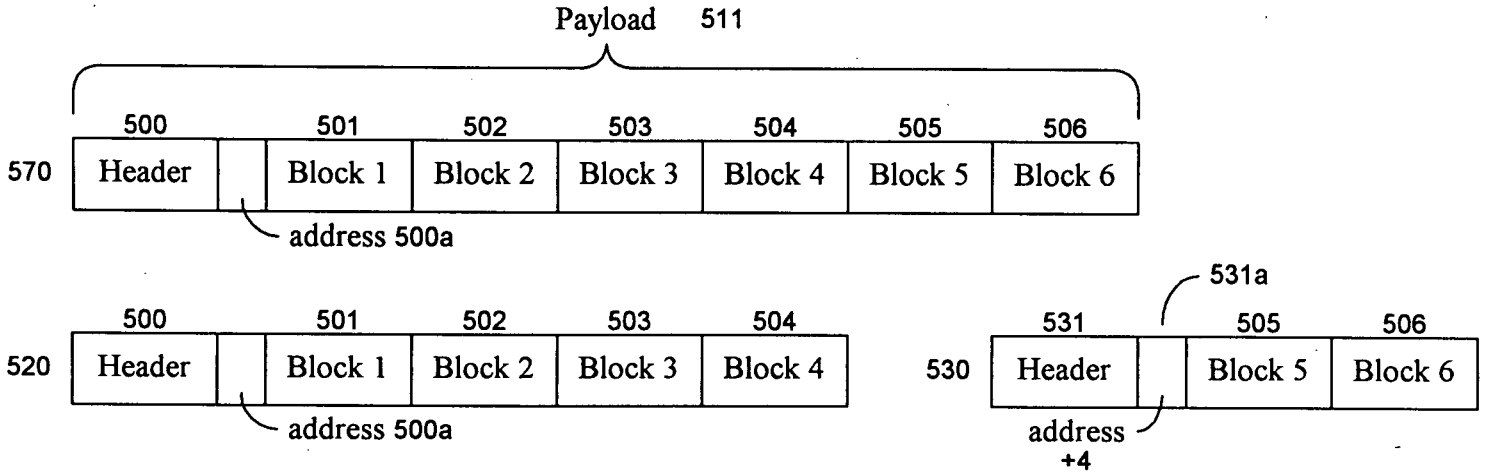

Payload 511
500
501
502
503
504
505
506
570
Header
Block 1
Block 2
Block 3
Block 4
Block 5
Block 6
address 500a
500
501
502
503
504
520
Header
Block 1
Block 2
Block 3
Block 4
address 500a
531a
531
505
506
530
Header
Block 5
Block 6
address
+4

Fig. 5

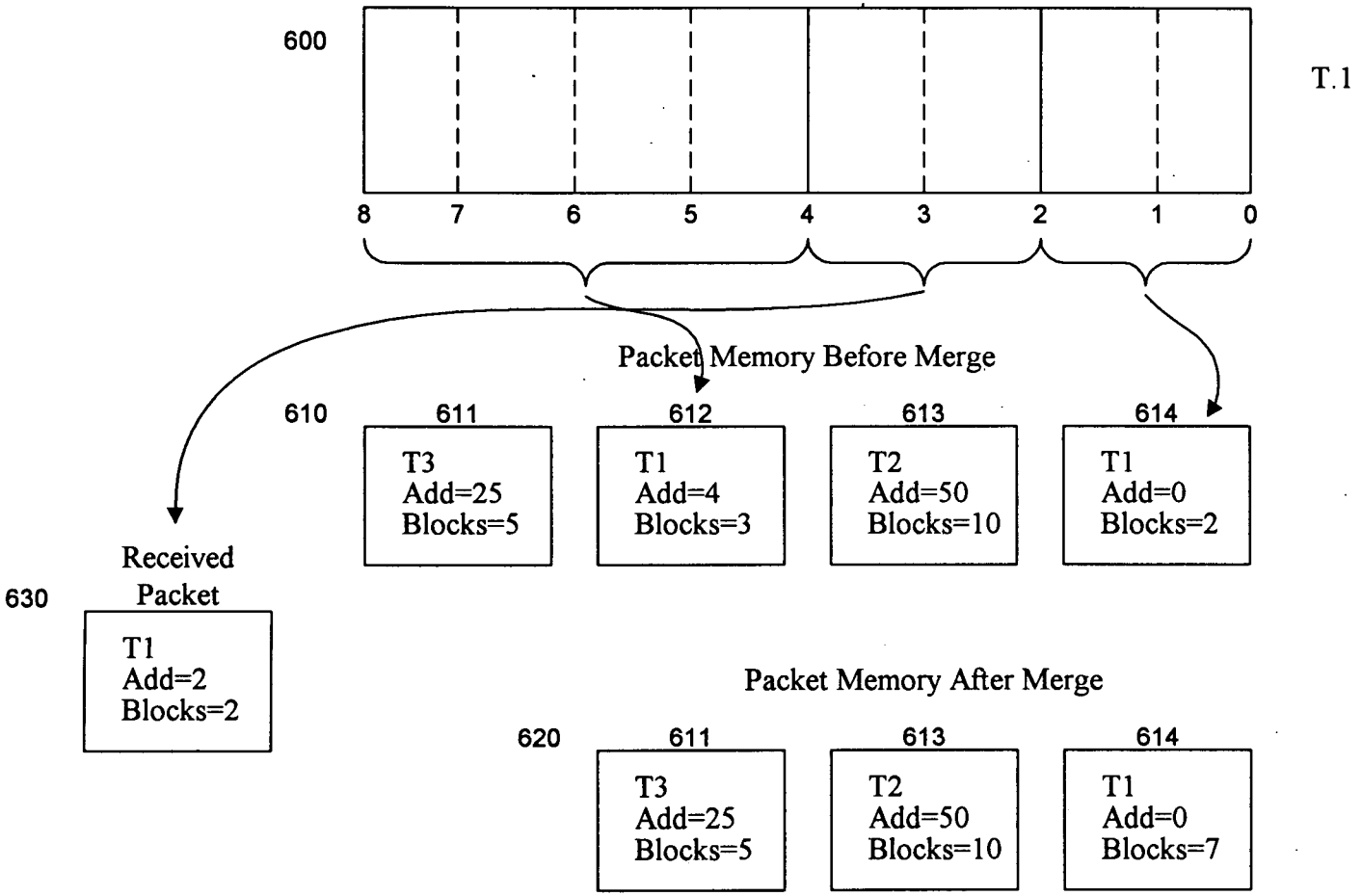


Fig. 6

1005345.05400

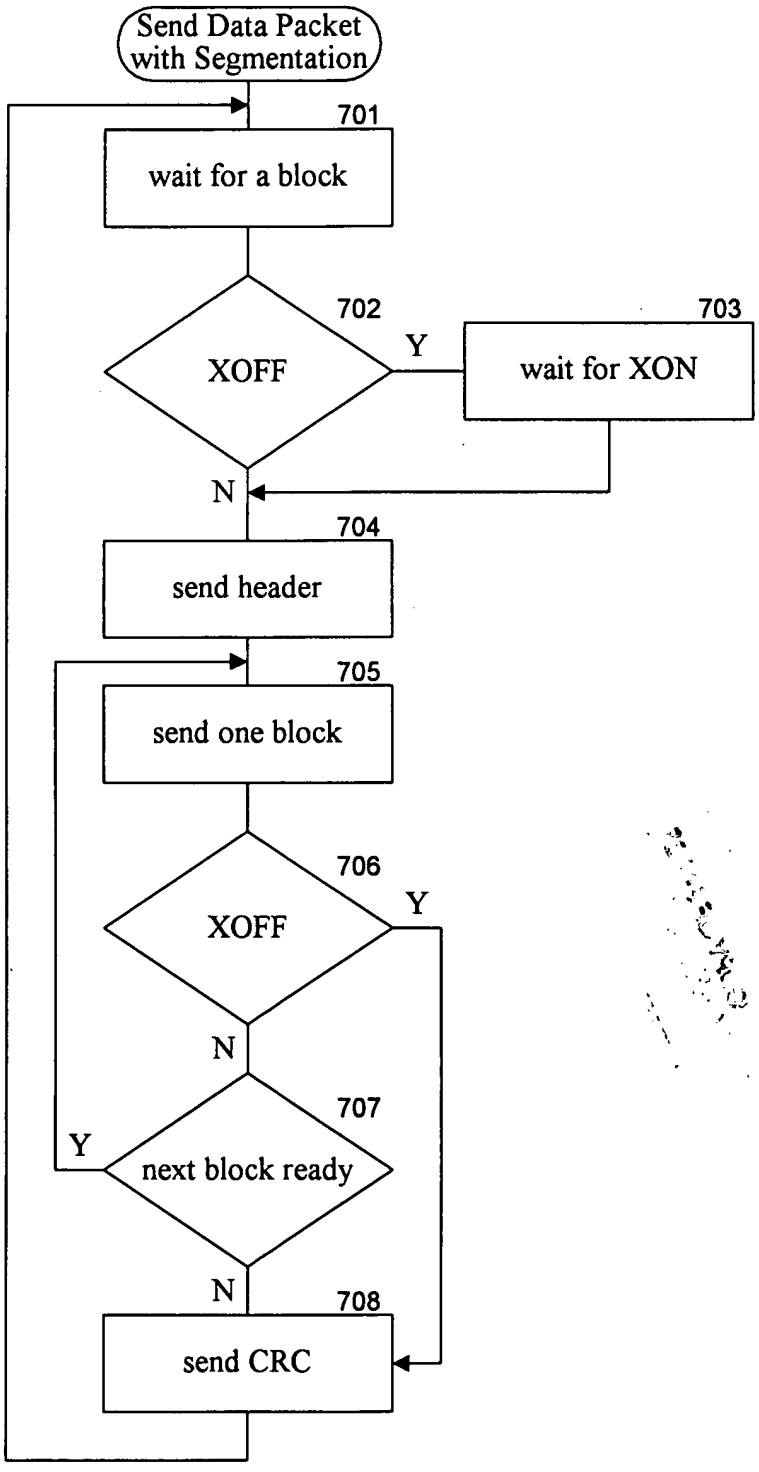

Send Data Packet with Segmentation
701
wait for a block
702
XOFF
Y
703
wait for XON
N
704
send header
705
send one block
706
XOFF
Y
N
707
next block ready
Y
N
708
send CRC

Fig. 7

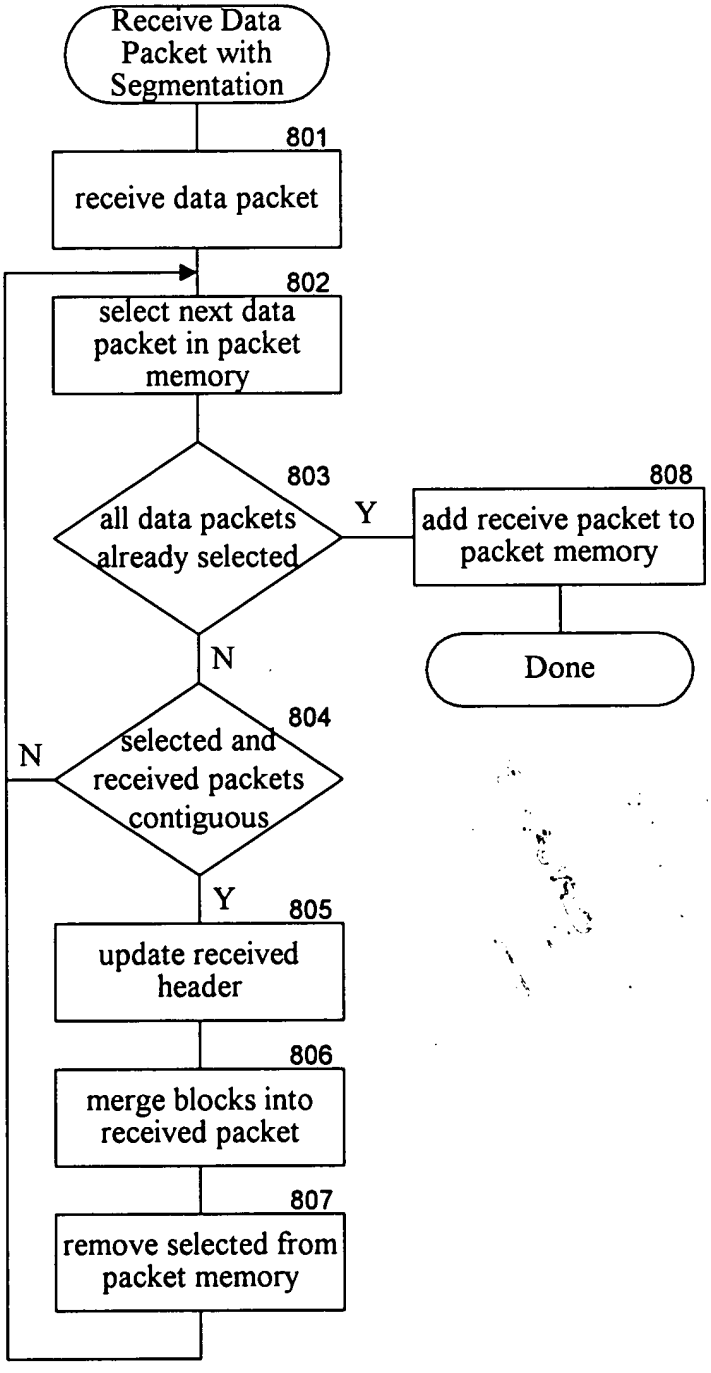
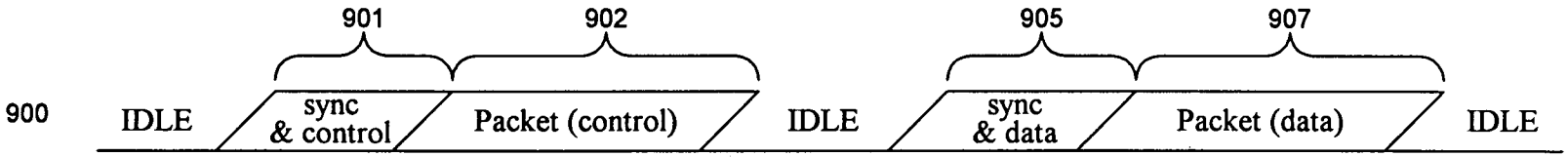

Receive Data Packet with Segmentation
801
receive data packet
802
select next data packet in packet memory
803
all data packets already selected
Y
808
add receive packet to packet memory
Done
N
804
selected and received packets contiguous
N
Y
805
update received header
806
merge blocks into received packet
807
remove selected from packet memory

Fig. 8



sync & packet type

Fig. 9A

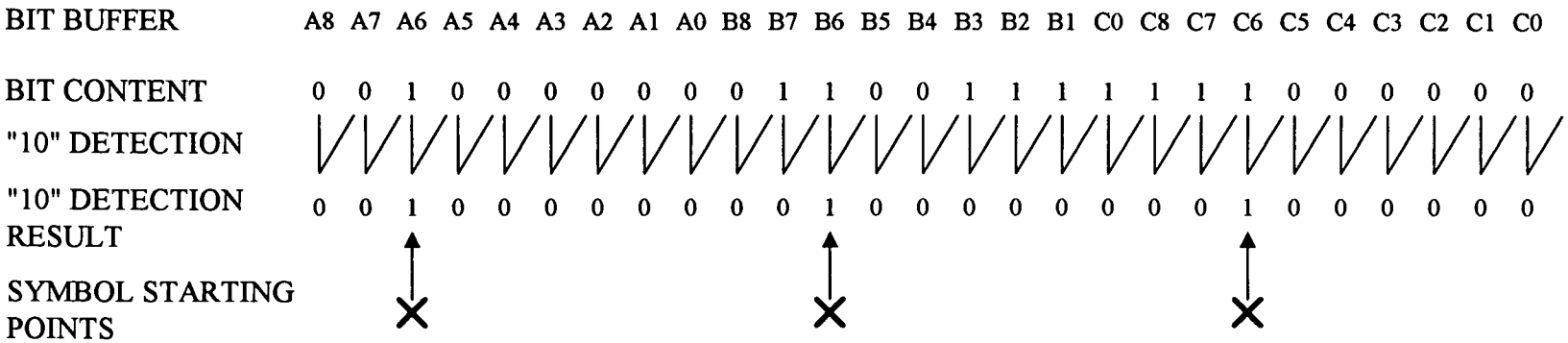


Fig. 9B

1005245 - 051002

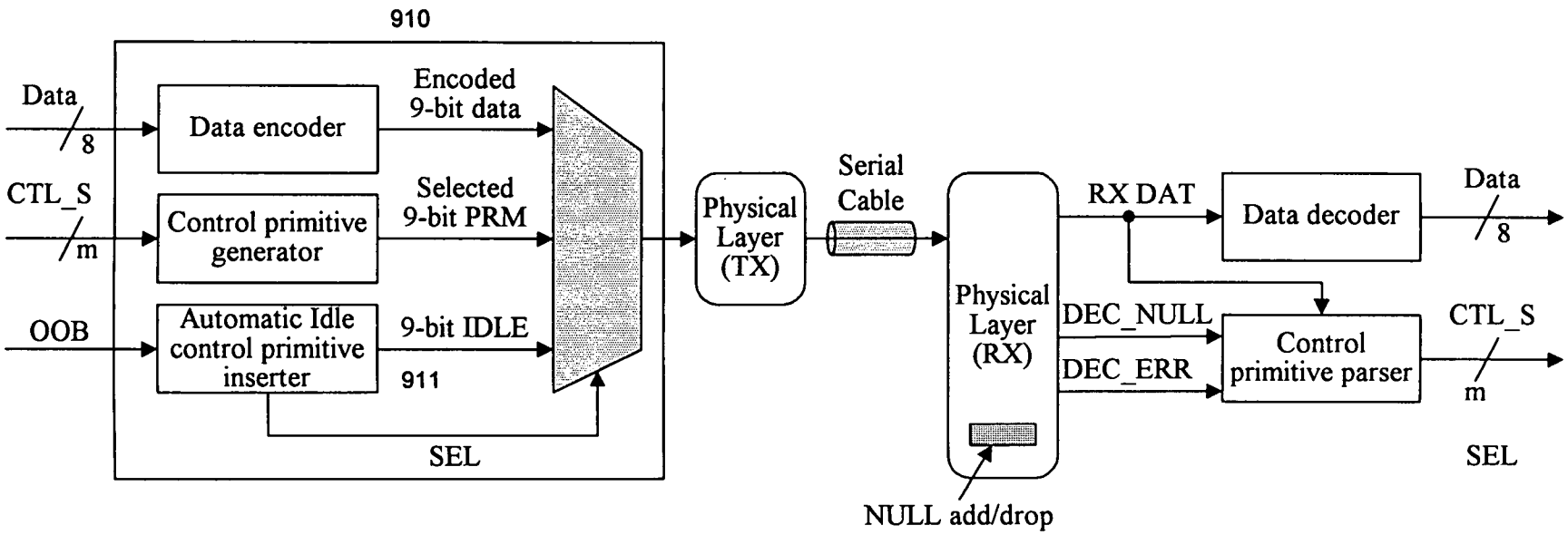


Fig. 9C

10052464.052002

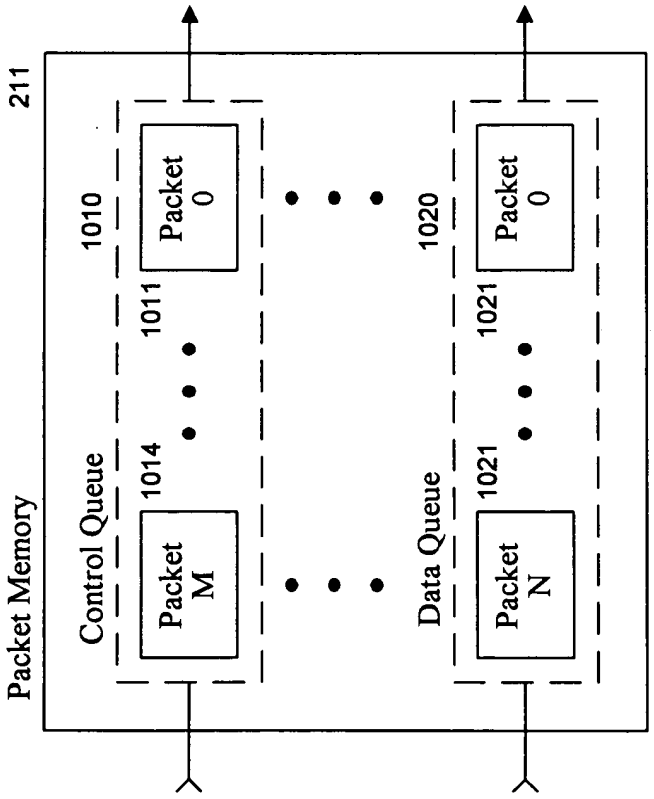

Packet Memory
211
Control Queue
1010
1011
Packet 0
1014
Packet M
Data Queue
1020
1021
Packet 0
1021
Packet N

Fig. 10

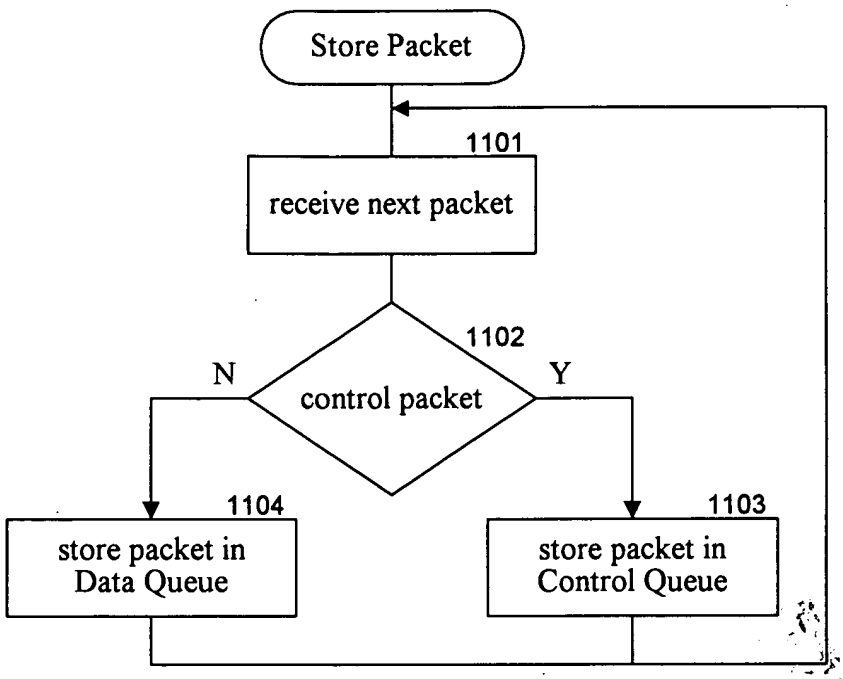

Store Packet
1101
receive next packet
1102
N
control packet
Y
1104
store packet in Data Queue
1103
store packet in Control Queue

Fig. 11

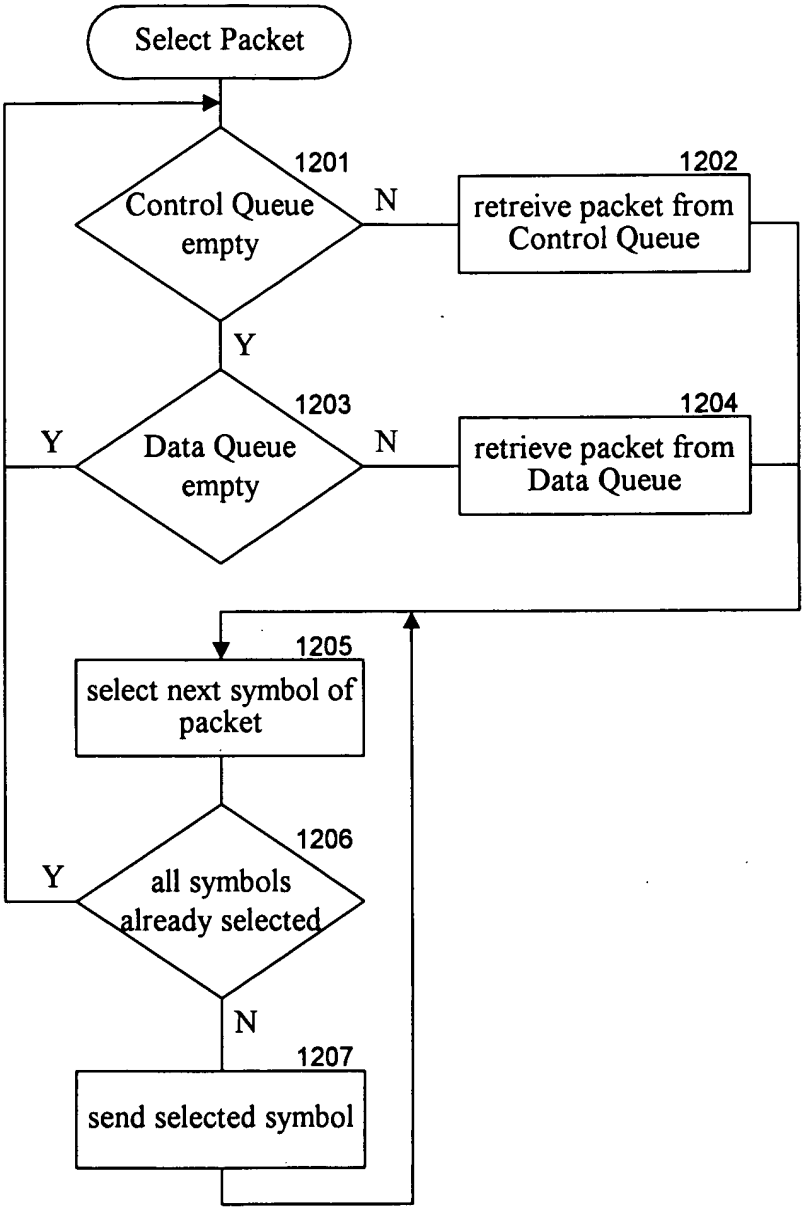

Select Packet
1201
Control Queue empty
N
1202
retreive packet from Control Queue
Y
1203
Data Queue empty
Y
N
1204
retrieve packet from Data Queue
1205
select next symbol of packet
1206
all symbols already selected
Y
N
1207
send selected symbol

Fig. 12

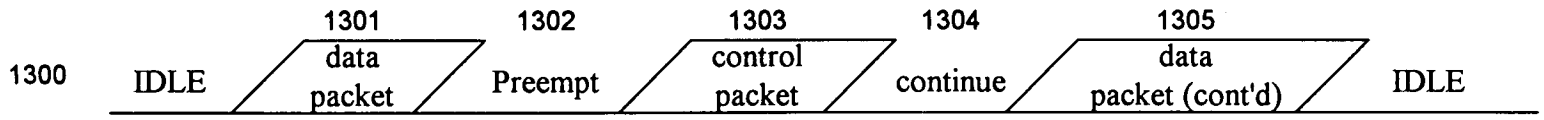

1300
IDLE
1301
data packet
1302
Preempt
1303
control packet
1304
continue
1305
data packet (cont'd)
IDLE

Fig. 13

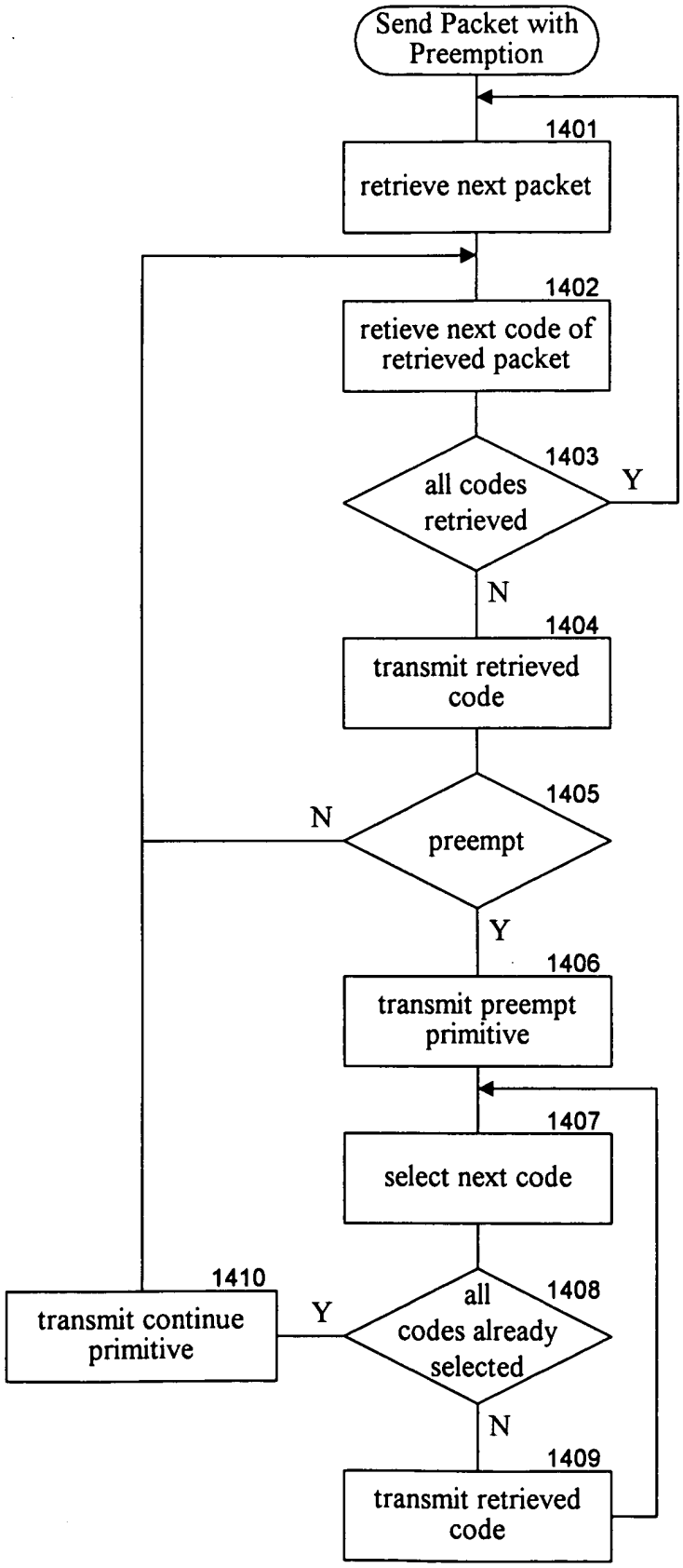

Send Packet with Preemption
1401
retrieve next packet
1402
retieve next code of retrieved packet
1403
all codes retrieved
Y
N
1404
transmit retrieved code
1405
preempt
N
Y
1406
transmit preempt primitive
1407
select next code
1408
all codes already selected
Y
N
1410
transmit continue primitive
1409
transmit retrieved code

Fig. 14

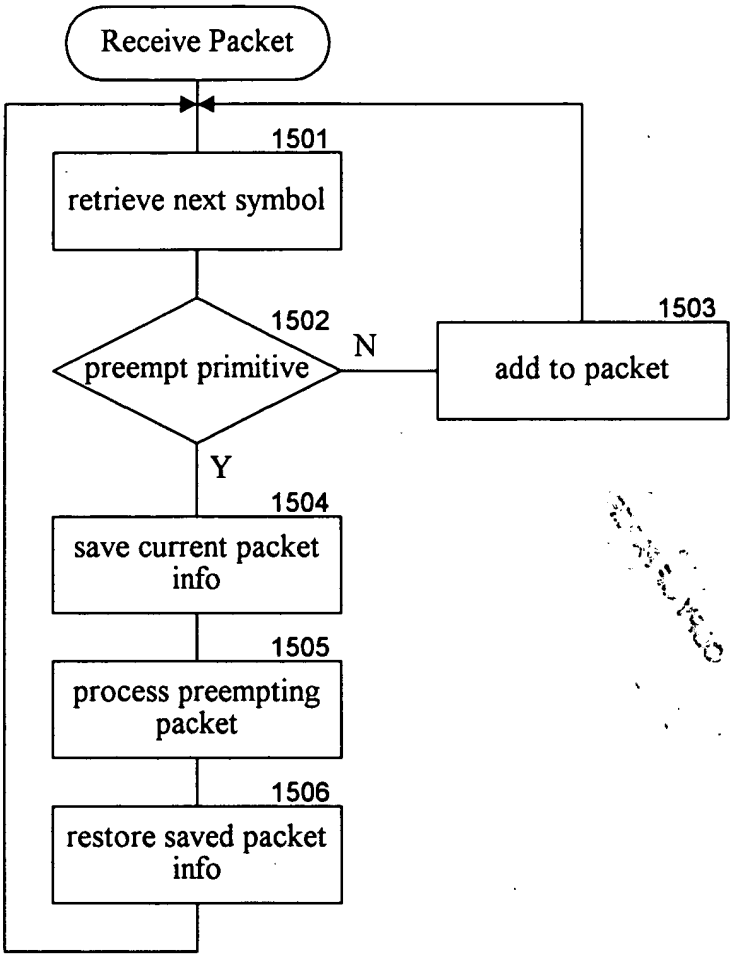

Receive Packet
1501
retrieve next symbol
1502
preempt primitive
N
1503
add to packet
Y
1504
save current packet info
1505
process preempting packet
1506
restore saved packet info

Fig. 15

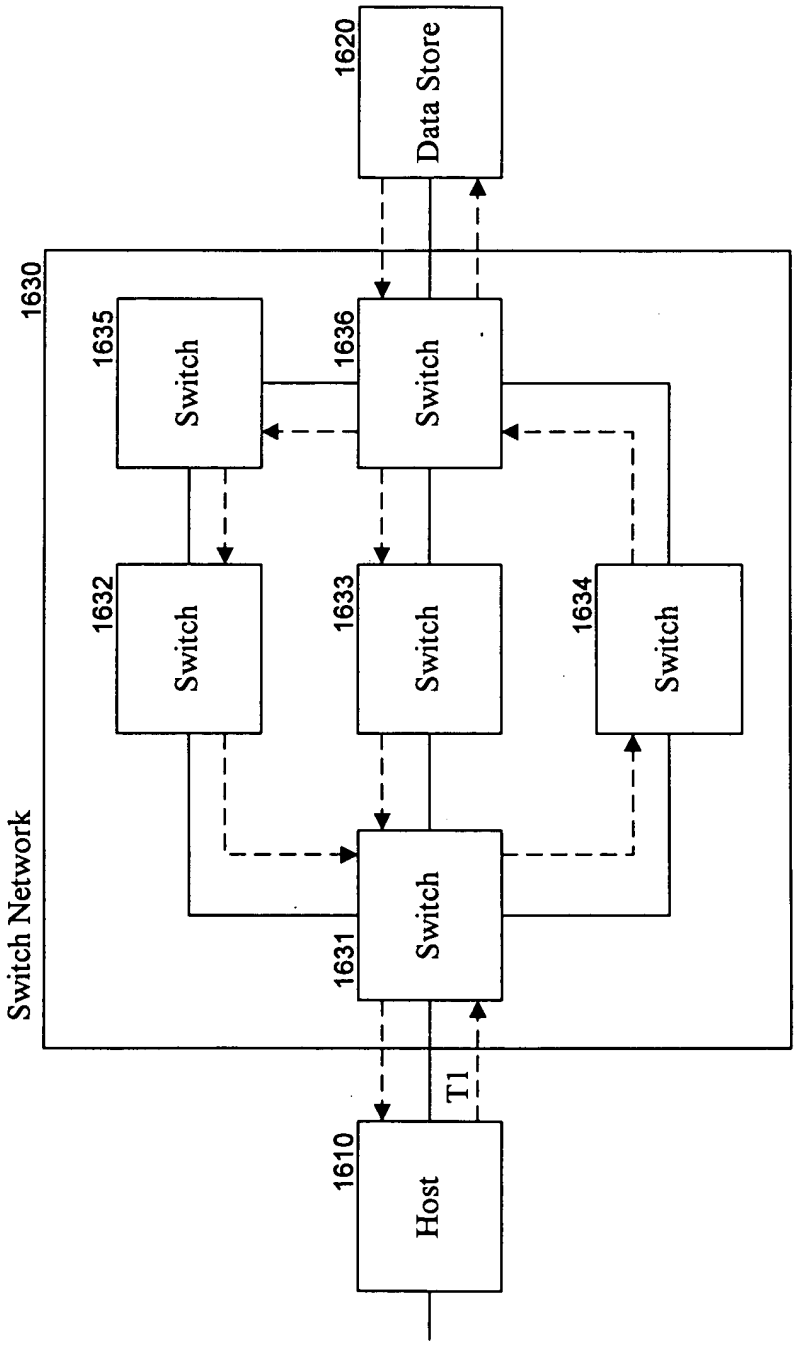

1620
Data Store
1630
Switch Network
1635
Switch
1636
Switch
1632
Switch
1633
Switch
1634
Switch
1631
Switch
T1
1610
Host

Fig. 16

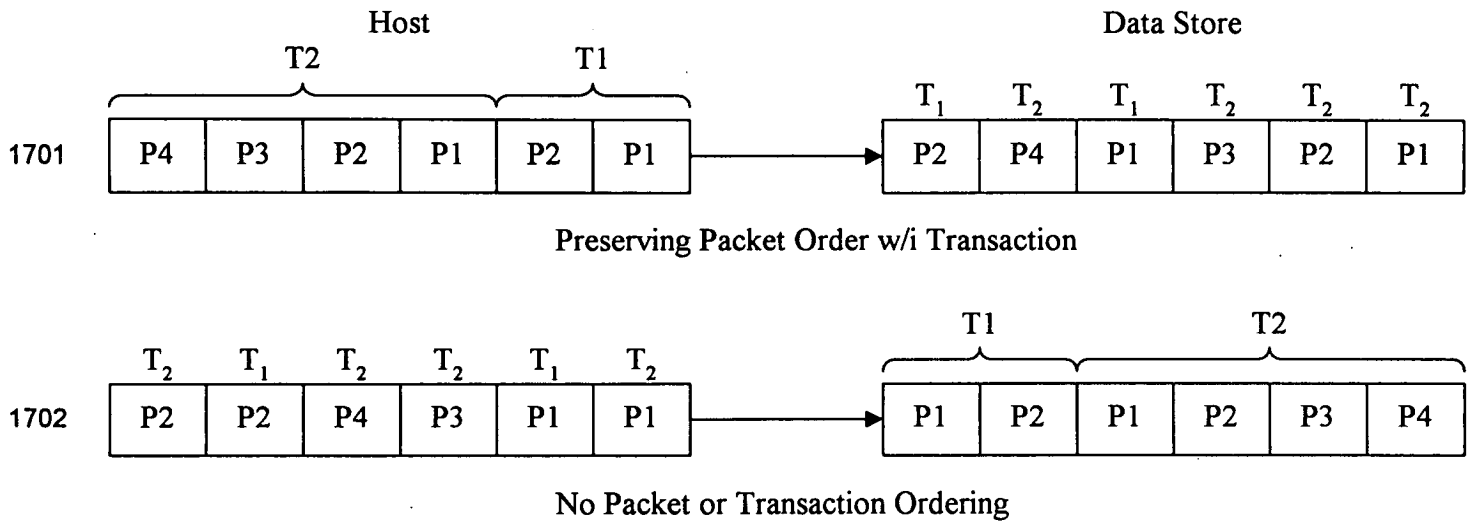


Fig. 17

10053462.051002

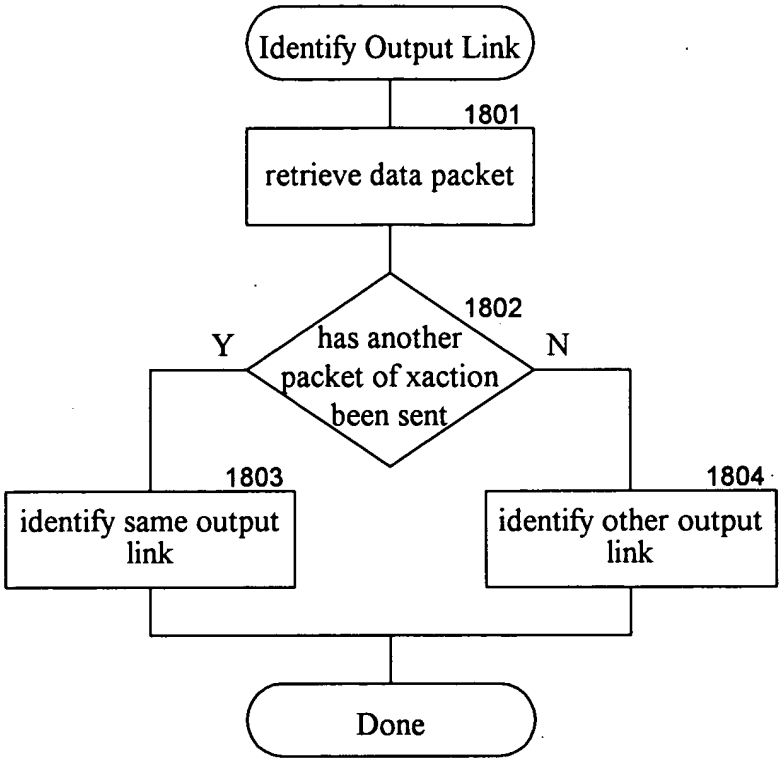

Identify Output Link
1801
retrieve data packet
1802
has another packet of xaction been sent
Y
N
1803
identify same output link
1804
identify other output link
Done

Fig. 18

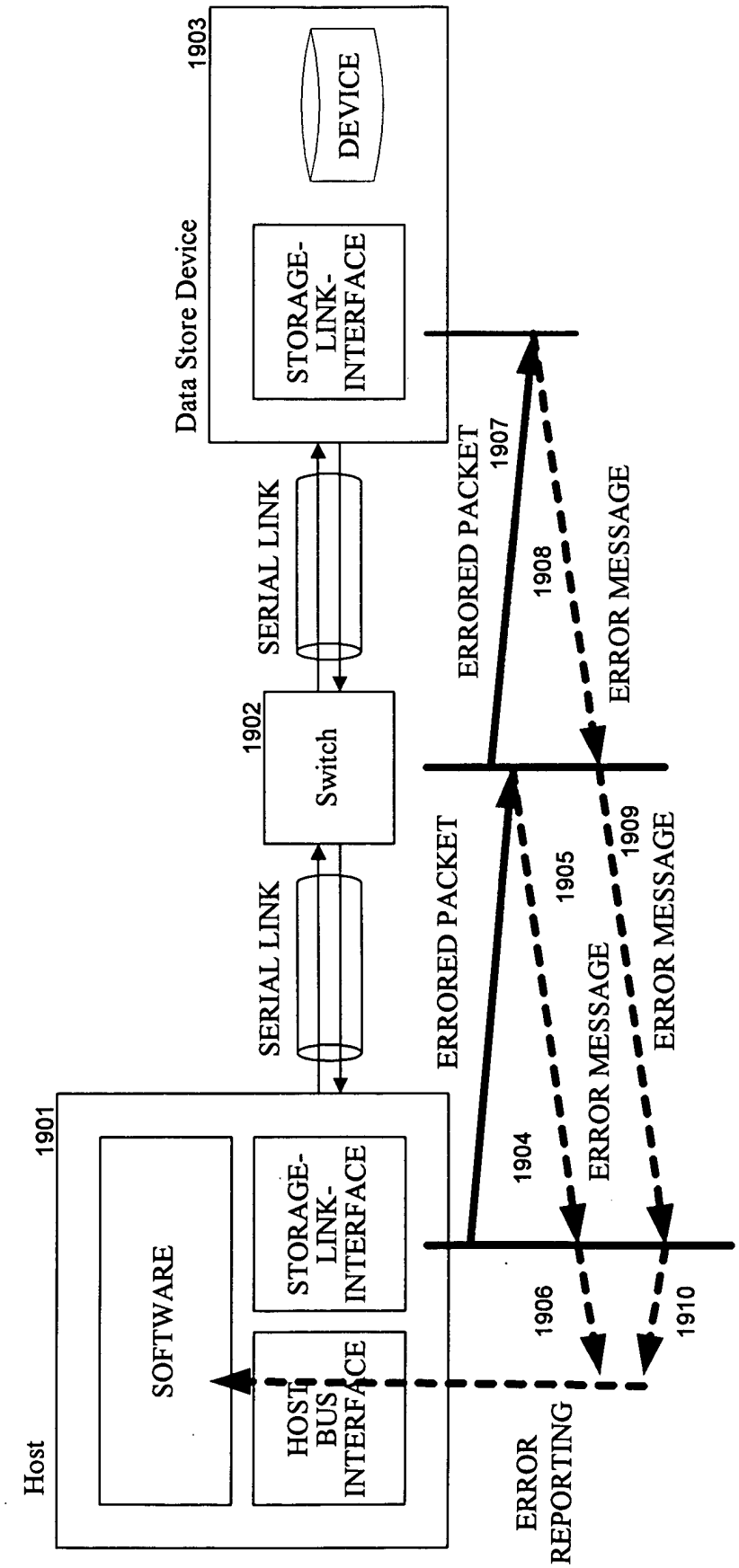

Host
1901
SOFTWARE
HOST BUS INTERFACE
STORAGE-LINK-INTERFACE
SERIAL LINK
1902
Switch
SERIAL LINK
Data Store Device
1903
STORAGE-LINK-INTERFACE
DEVICE
ERRORED PACKET
1904
ERROR MESSAGE
1905
ERRORED PACKET
1907
ERROR MESSAGE
1908
ERROR MESSAGE
1909
ERROR REPORTING
1906
1910

Fig. 19A

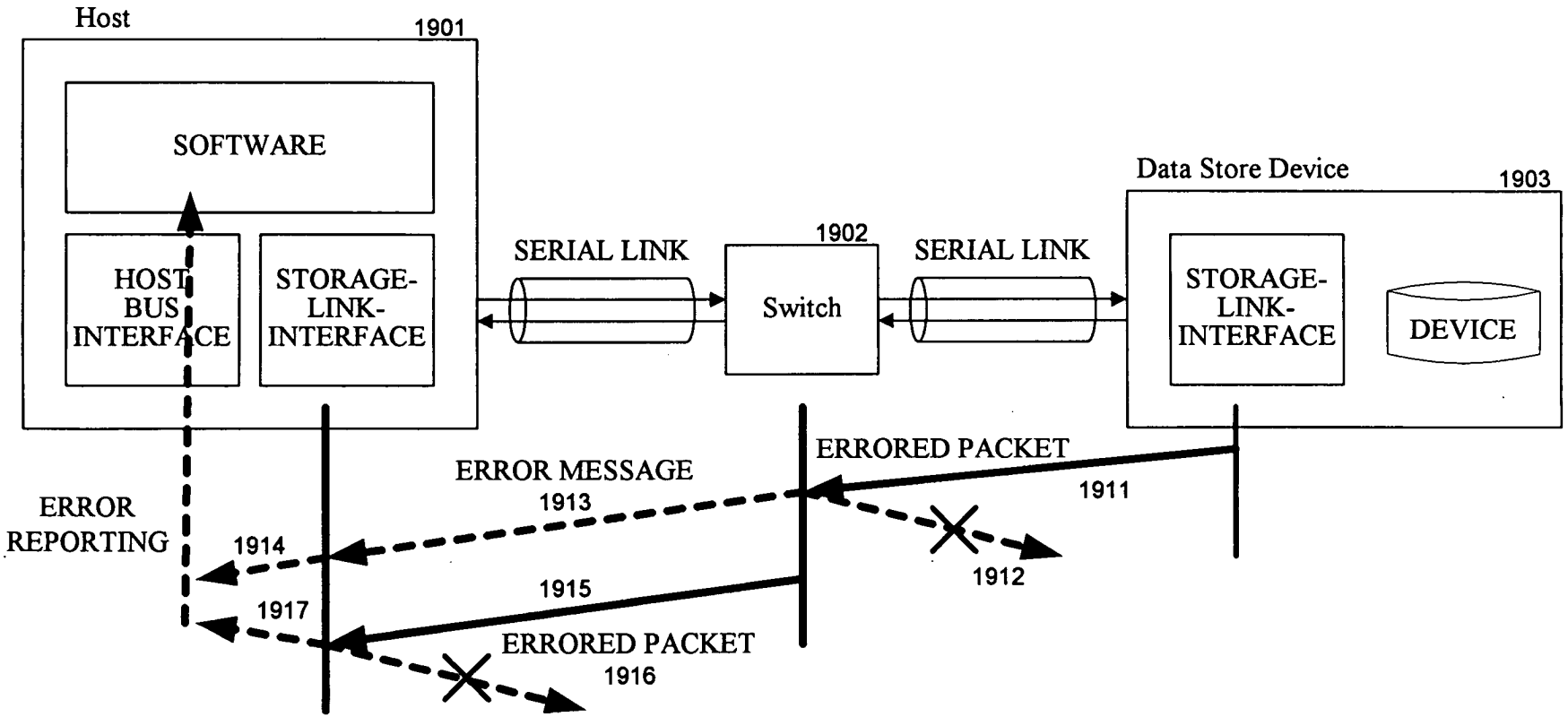


Fig. 19B

EXHIBIT 1005, PAGE 0054

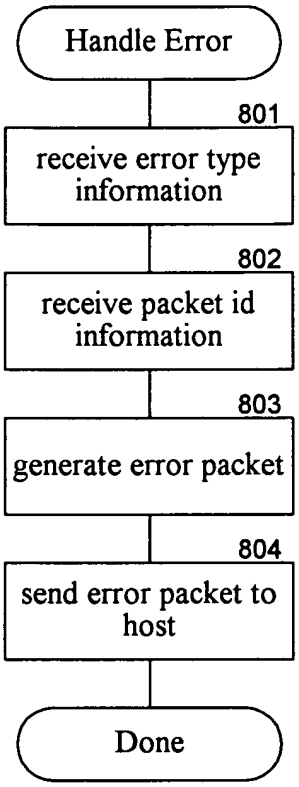

Handle Error
801
receive error type information
802
receive packet id information
803
generate error packet
804
send error packet to host
Done

Fig. 19C

8b code	9 bit symbol
0000 0000	101010101
0000 0001	101010100
0000 0010	101010111
⋮	
0101 0101	001010101
⋮	
0111 0110	001110110
0111 0111	100100010
⋮	
1111 1111	110101010

Fig. 20

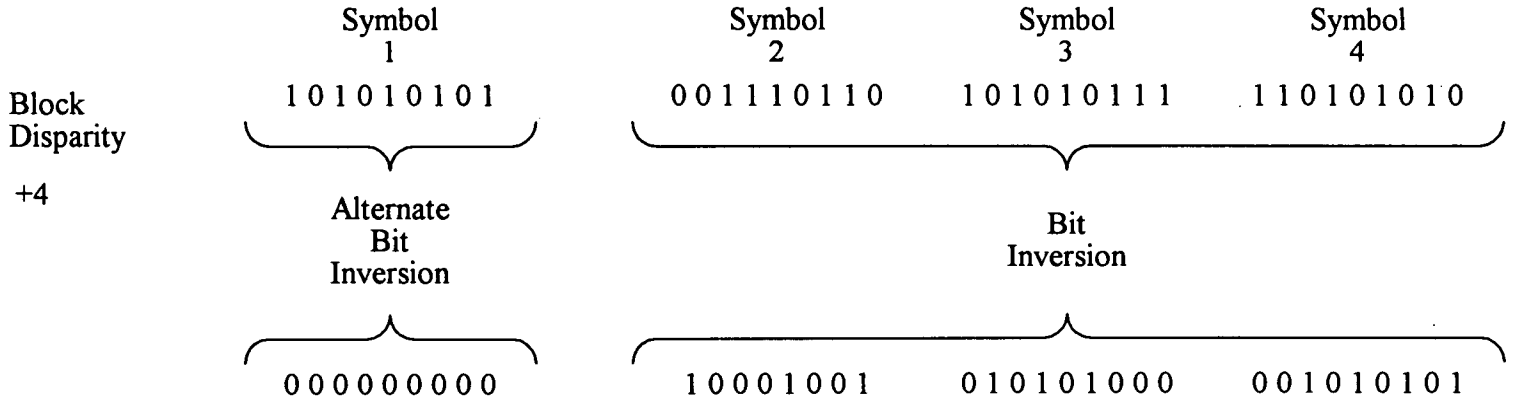


Fig. 21A

1005346.1 05-10-02

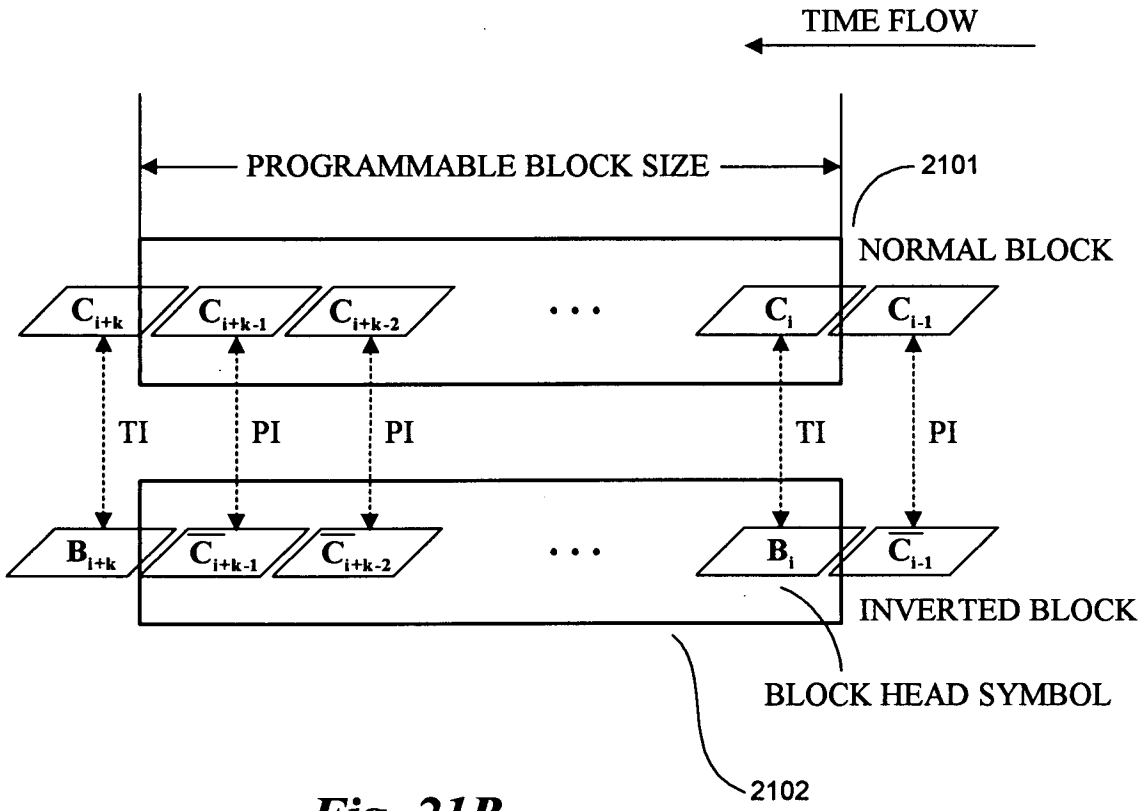


Fig. 21B

10053461 - 052102

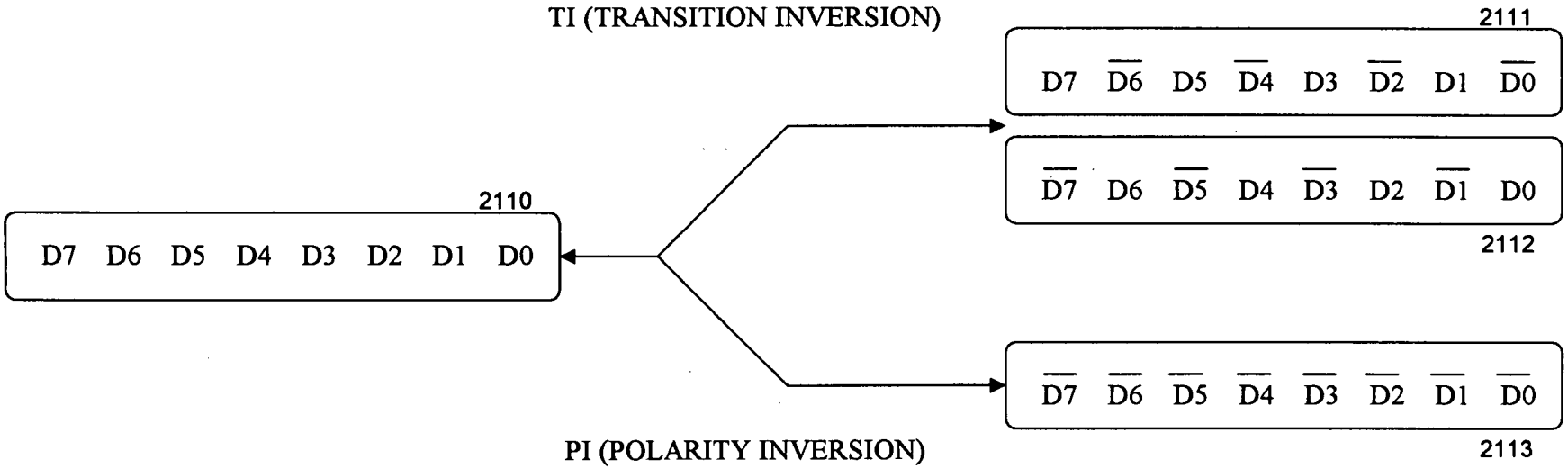


Fig. 21C

1005346.05.1002

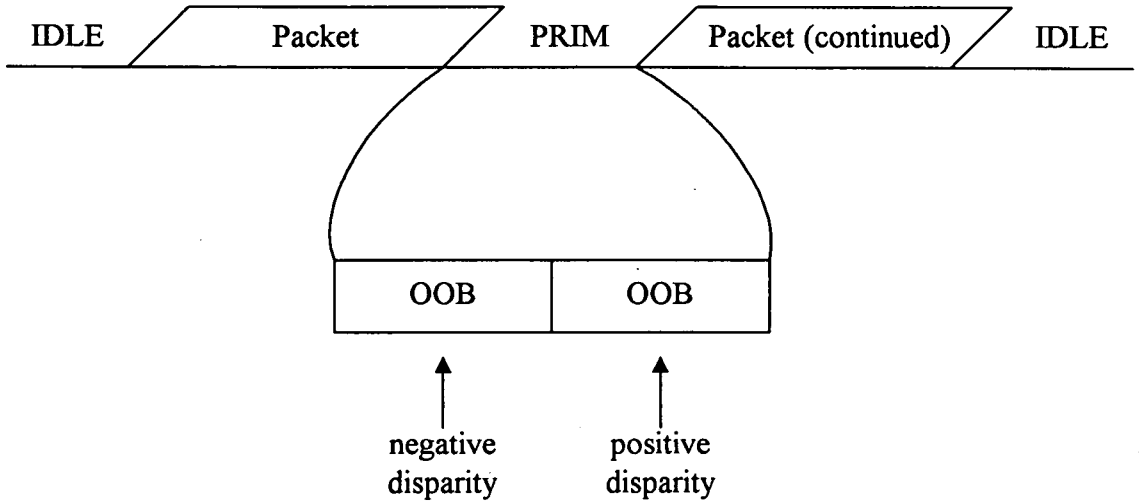


Fig. 22

10053461.050102

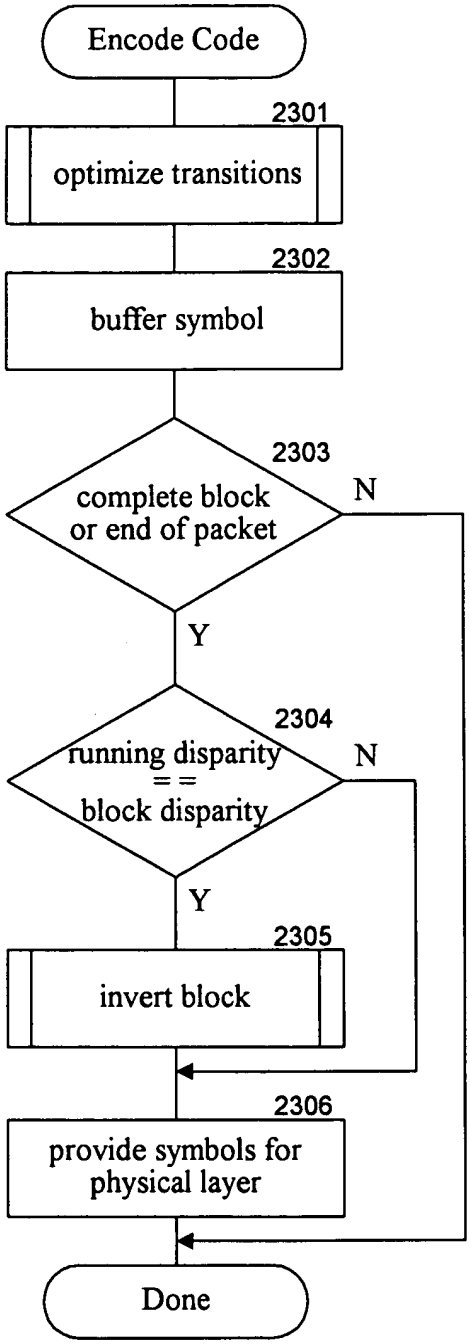

Encode Code
2301
optimize transitions
2302
buffer symbol
2303
complete block or end of packet
N
Y
2304
running disparity == block disparity
N
Y
2305
invert block
2306
provide symbols for physical layer
Done

Fig. 23

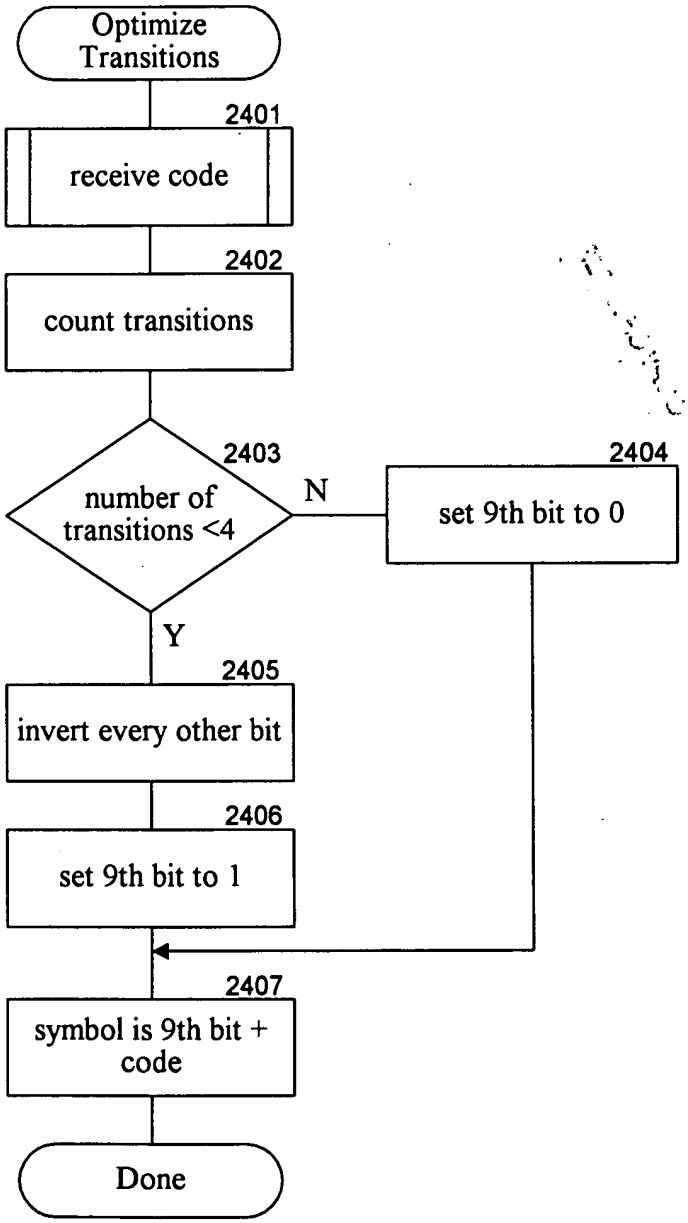

Optimize Transitions
2401
receive code
2402
count transitions
2403
number of transitions <4
N
2404
set 9th bit to 0
Y
2405
invert every other bit
2406
set 9th bit to 1
2407
symbol is 9th bit + code
Done

Fig. 24

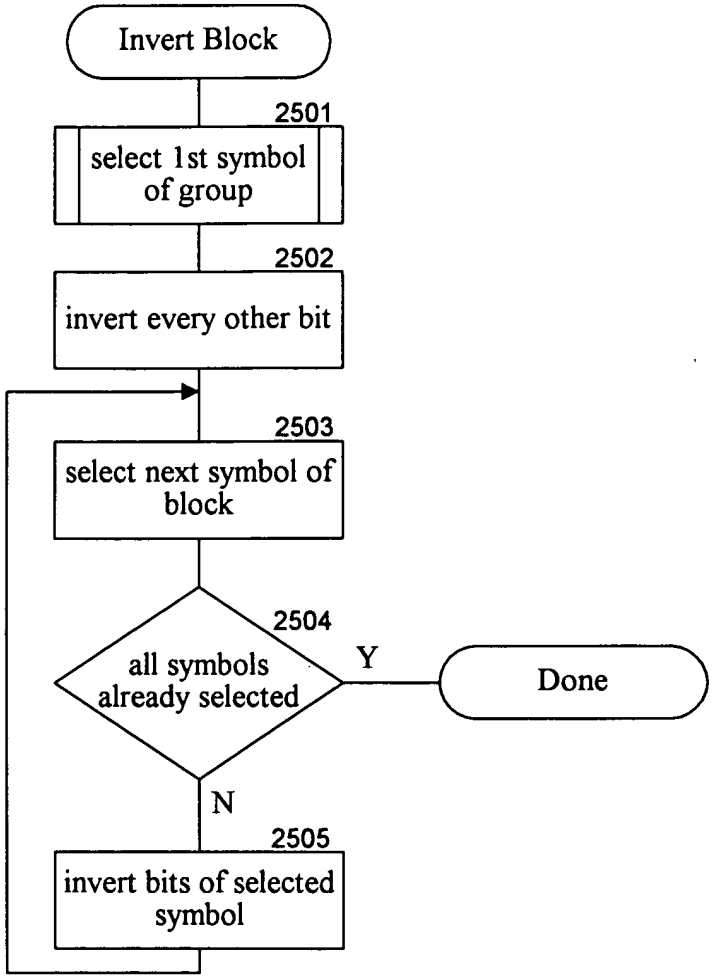

Invert Block
2501
select 1st symbol of group
2502
invert every other bit
2503
select next symbol of block
2504
all symbols already selected
Y
Done
N
2505
invert bits of selected symbol

Fig. 25

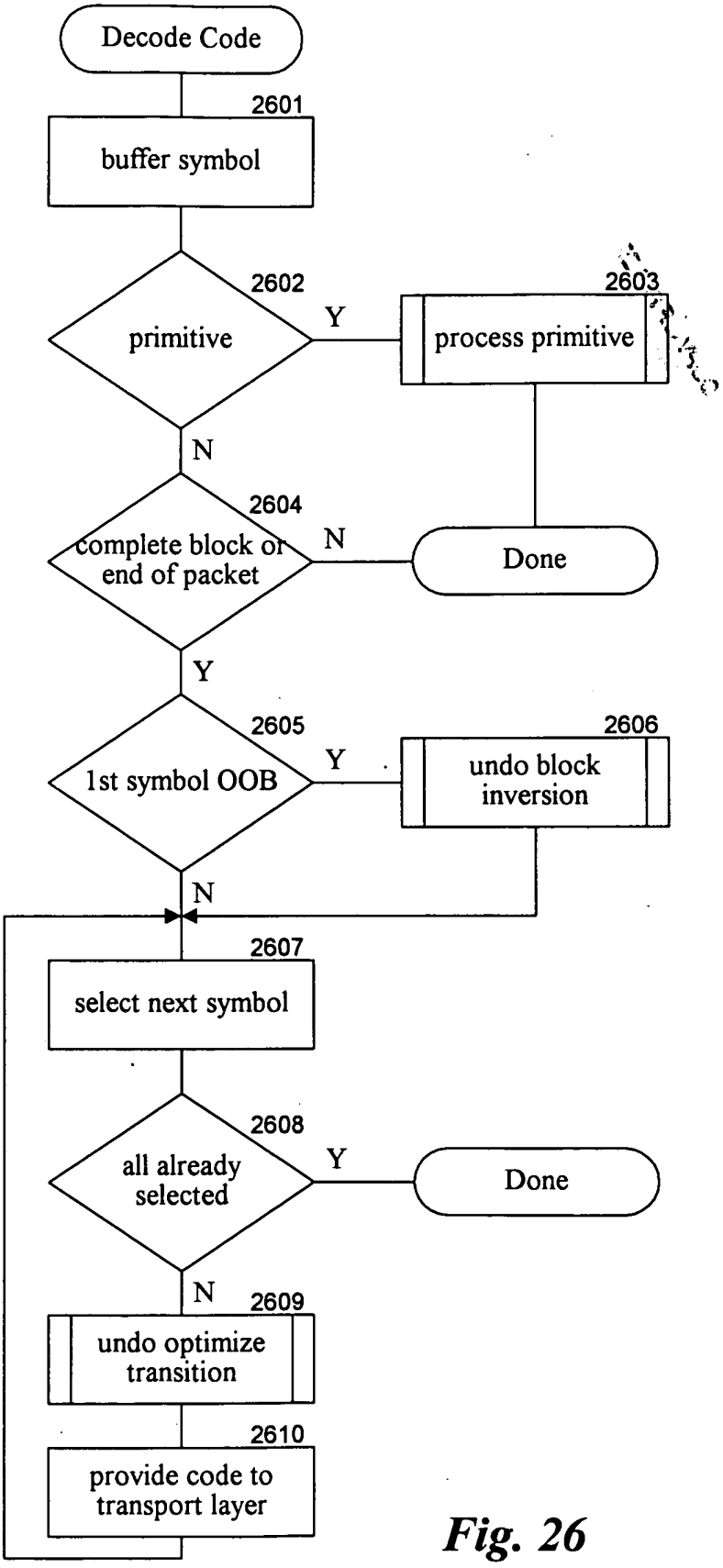

Decode Code
2601
buffer symbol
2602
primitive
Y
2603
process primitive
N
2604
complete block or end of packet
N
Done
Y
2605
1st symbol OOB
Y
2606
undo block inversion
N
2607
select next symbol
2608
all already selected
Y
Done
N
2609
undo optimize transition
2610
provide code to transport layer

Fig. 26

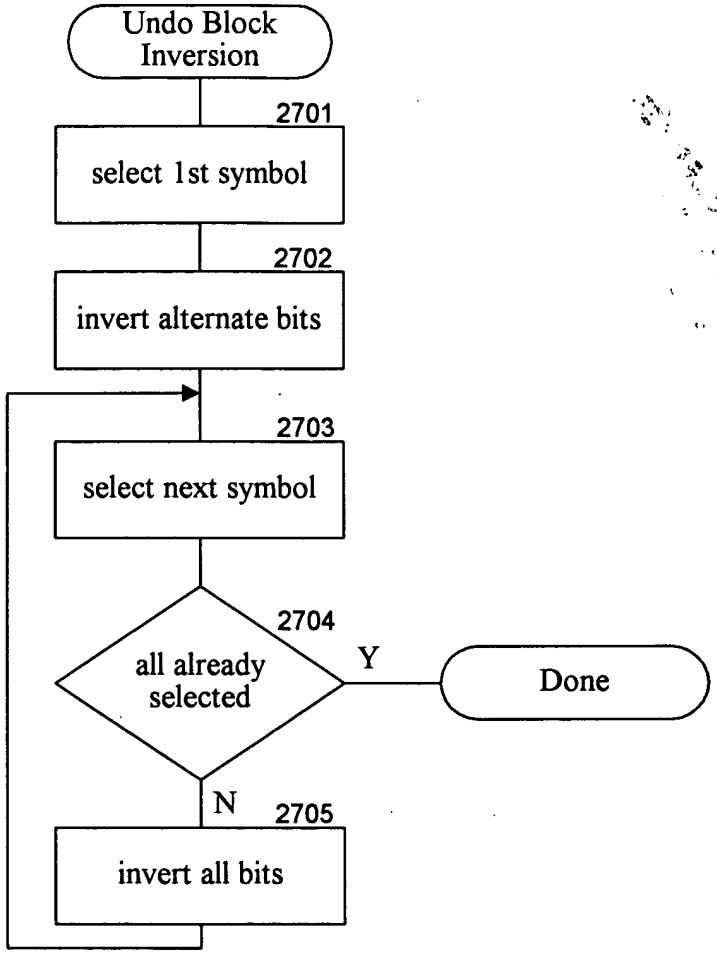

Undo Block Inversion
2701
select 1st symbol
2702
invert alternate bits
2703
select next symbol
2704
all already selected
Y
Done
N
2705
invert all bits

Fig. 27

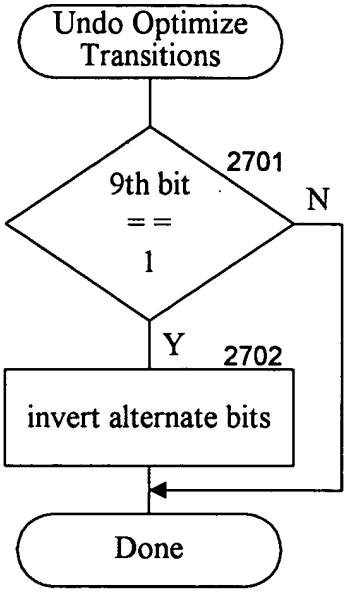

Undo Optimize Transitions
9th bit == 1
2701
N
Y
2702
invert alternate bits
Done

Fig. 28

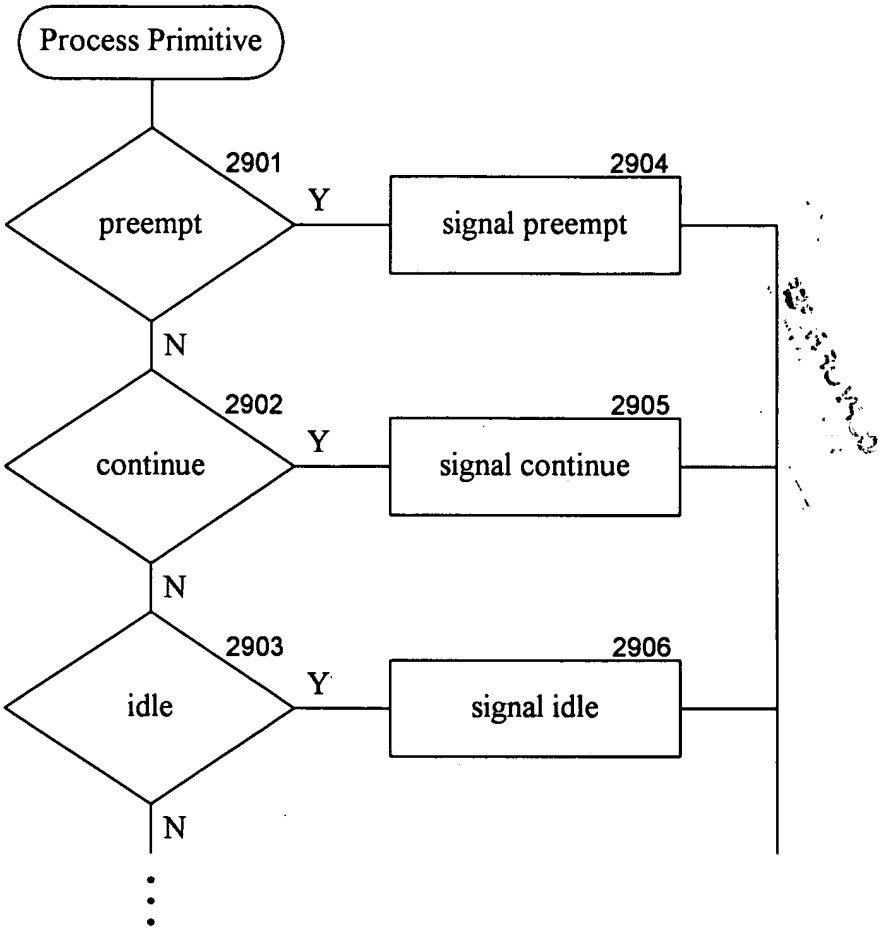

Process Primitive
2901
preempt
Y
2904
signal preempt
N
2902
continue
Y
2905
signal continue
N
2903
idle
Y
2906
signal idle
N

Fig. 29

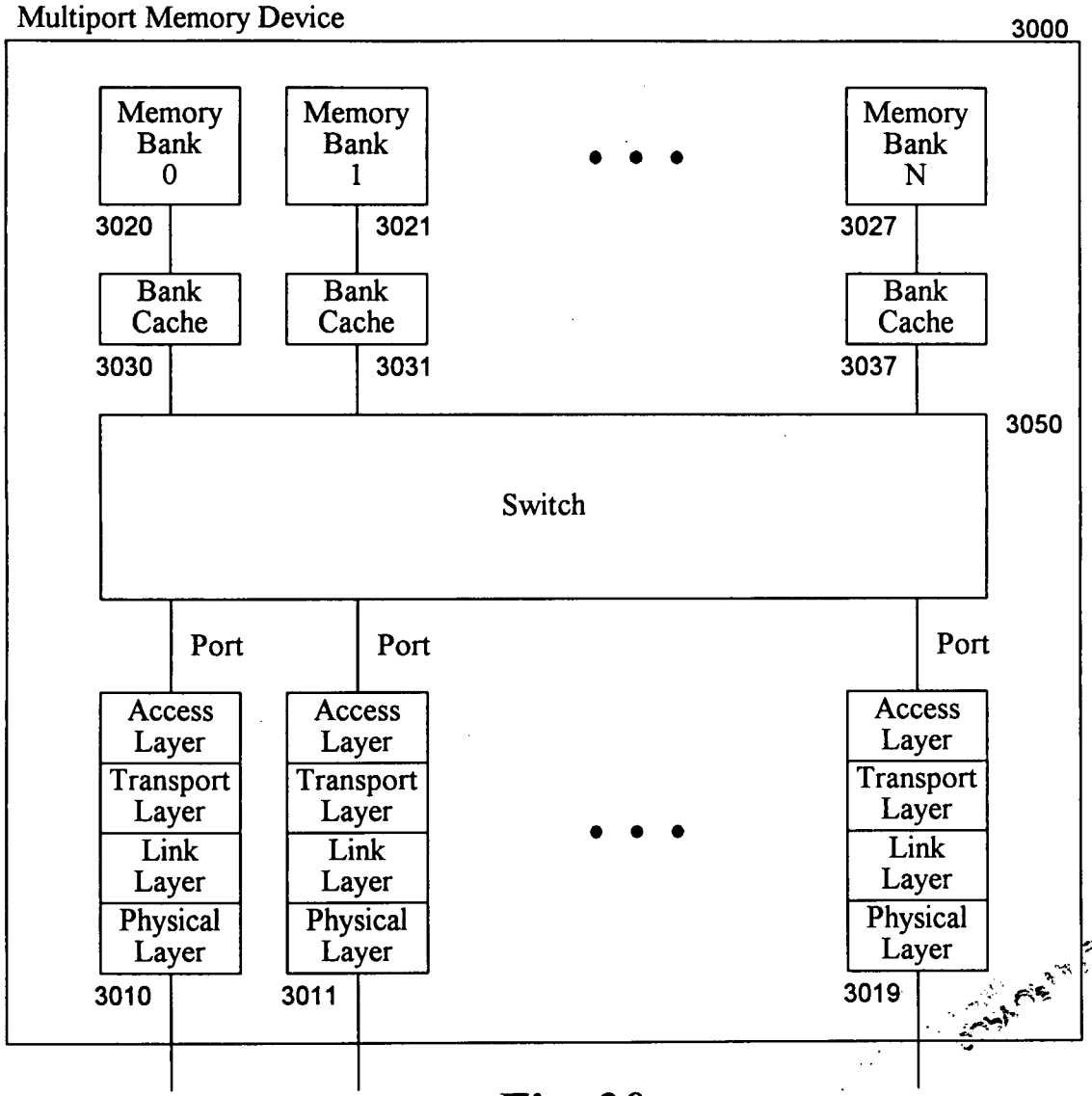


Fig. 30

1005346 "05100

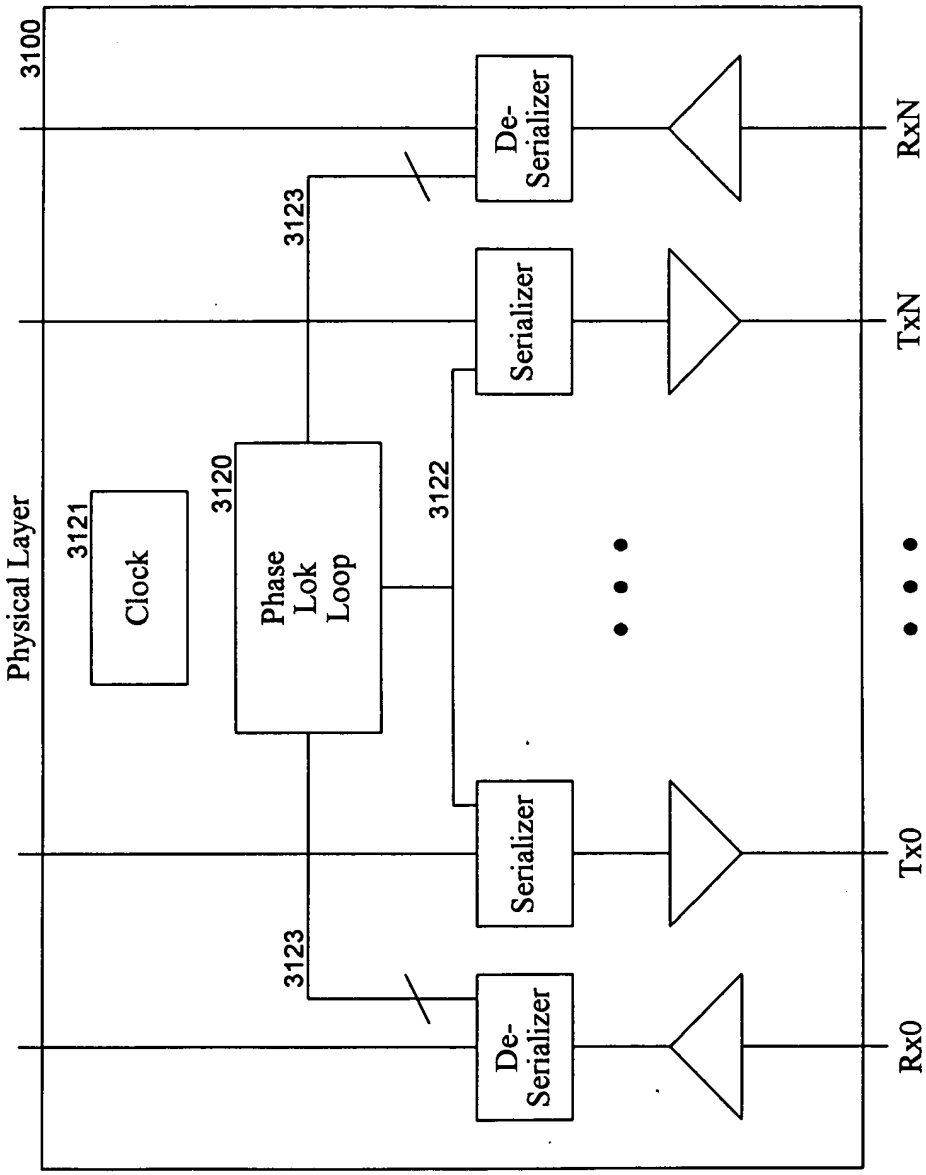

3100
Physical Layer
3121
Clock
3120
Phase
Lok
Loop
3122
3123
3123
De-
Serializer
Serializer
Serializer
De-
Serializer
RxN
TxN
Tx0
Rx0

Fig. 31

Input Queue 3201

Port	R/W	Address	Data
3	R	1000	
4	W	4000	10....1
3	W	1000	111...0
3	R	2000	
	⋮		

Output Queue 3202

Valid	Port	Data
1	3	11...0
0		
0		
1	3	101...1
	⋮	

Fig. 32

1005345.05100

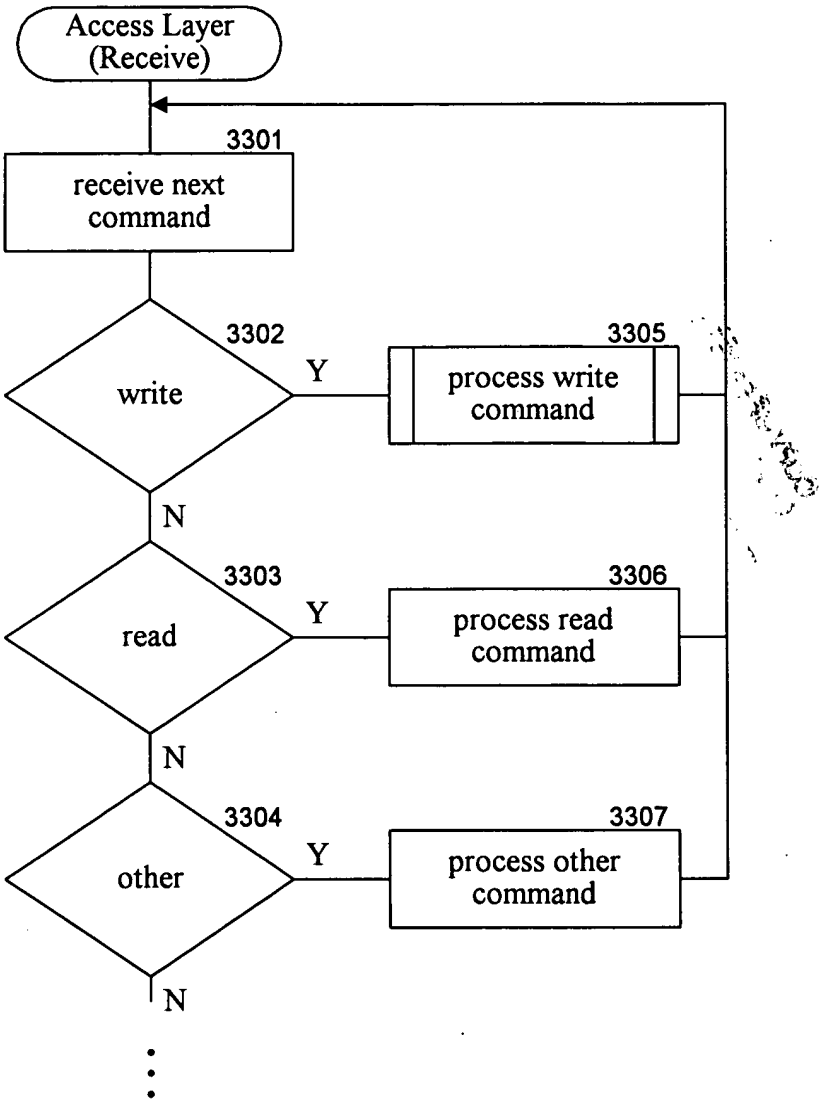

Access Layer
(Receive)
3301
receive next
command
3302
write
Y
3305
process write
command
N
3303
read
Y
3306
process read
command
N
3304
other
Y
3307
process other
command
N

Fig. 33

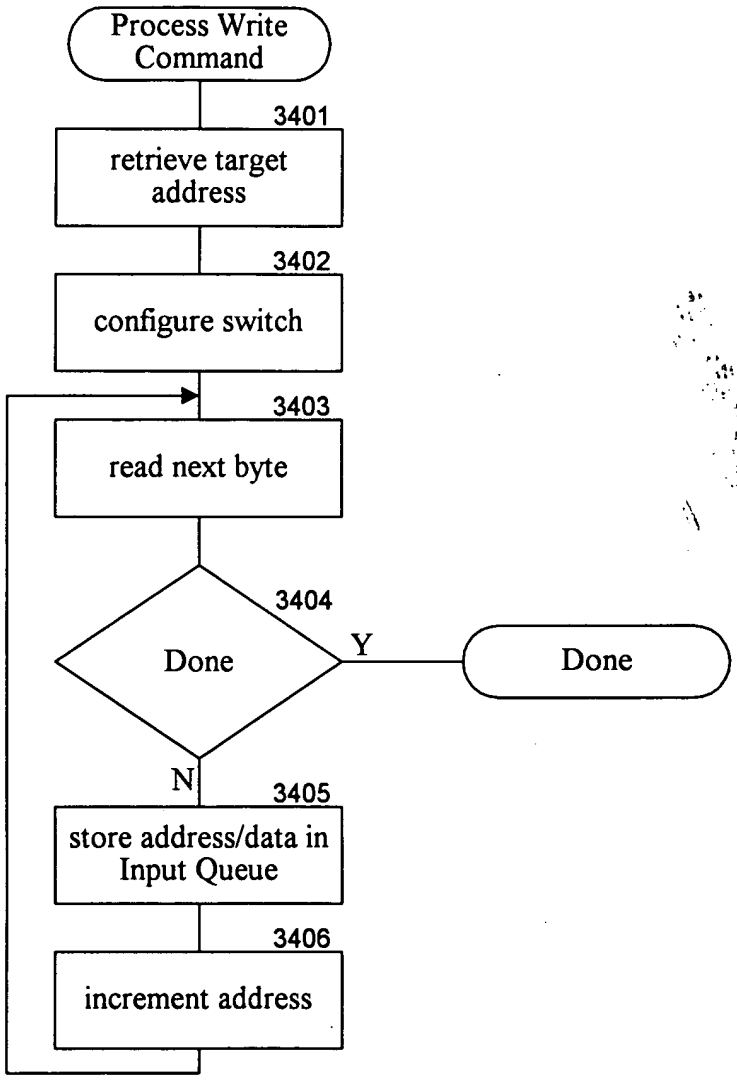

Process Write Command
3401
retrieve target address
3402
configure switch
3403
read next byte
3404
Done
Y
Done
N
3405
store address/data in Input Queue
3406
increment address

Fig. 34

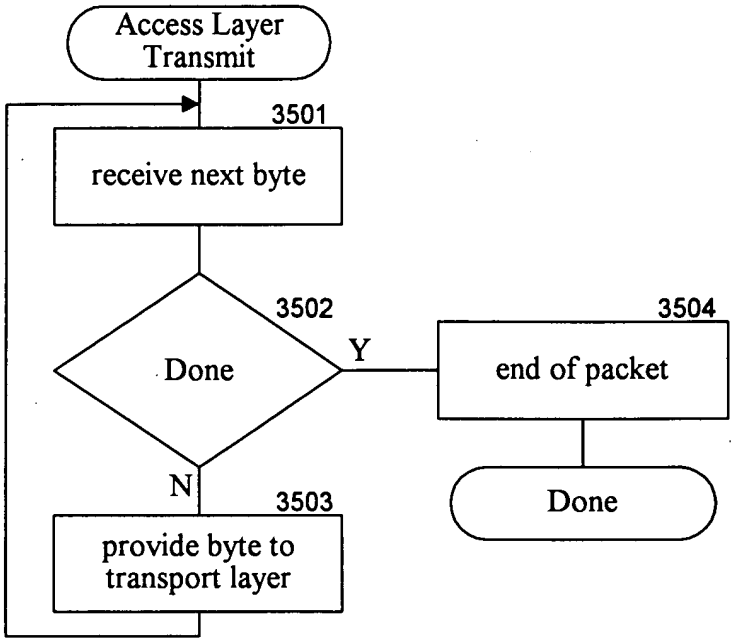

Access Layer Transmit
3501
receive next byte
3502
Done
Y
3504
end of packet
N
3503
provide byte to transport layer
Done

Fig. 35

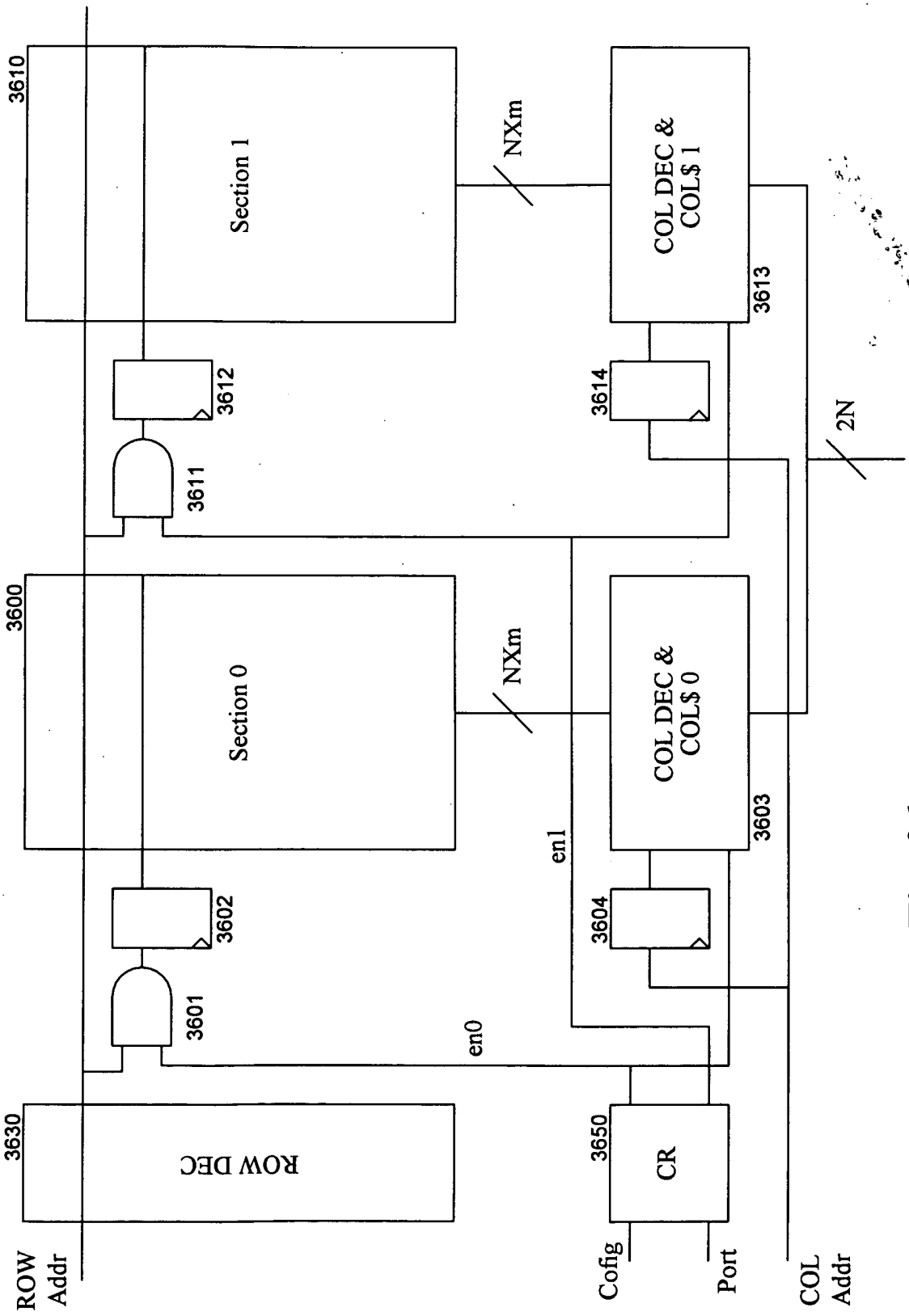
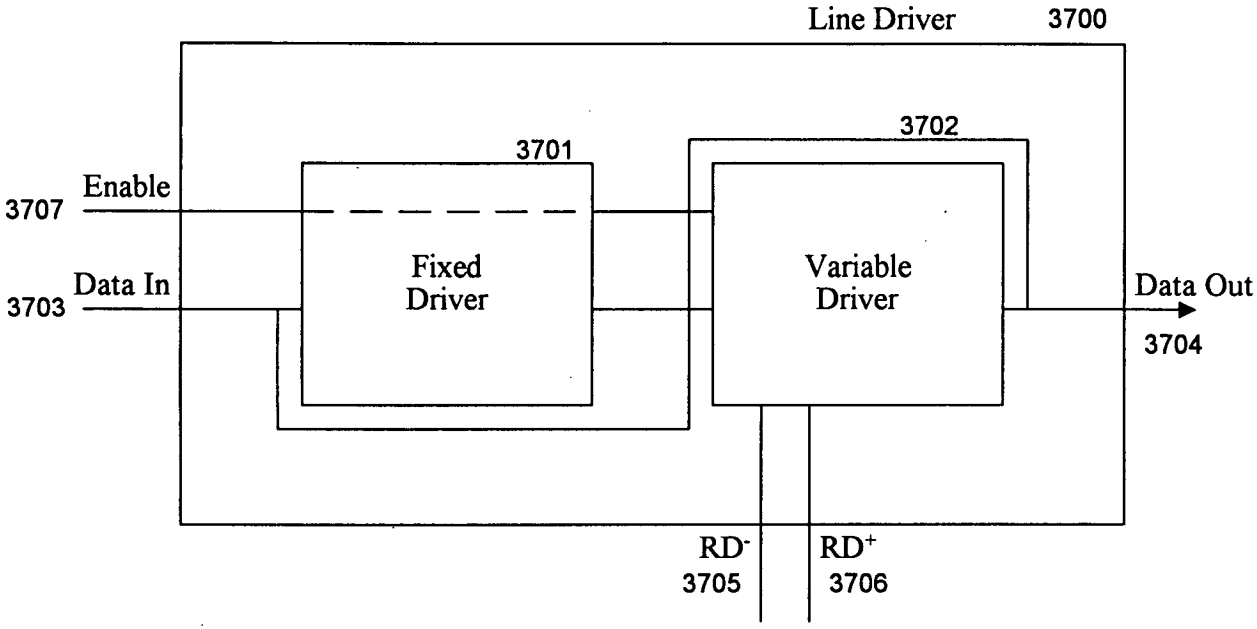

3610
Section 1
NXm
COL DEC &
COL$ 1
3613
3612
3614
3611
2N
3600
Section 0
NXm
COL DEC &
COL$ 0
3603
en1
3602
3604
3601
en0
3630
ROW DEC
3650
CR
ROW
Addr
Config
Port
COL
Addr

Fig. 36



Variable Driver
$\begin{cases} RD^+ \wedge \overline{DataIn} = \text{pull down} \\ RD^- \wedge DataIn = \text{pull up} \end{cases}$

Fig. 37A

1005346.1 - 05-10-02

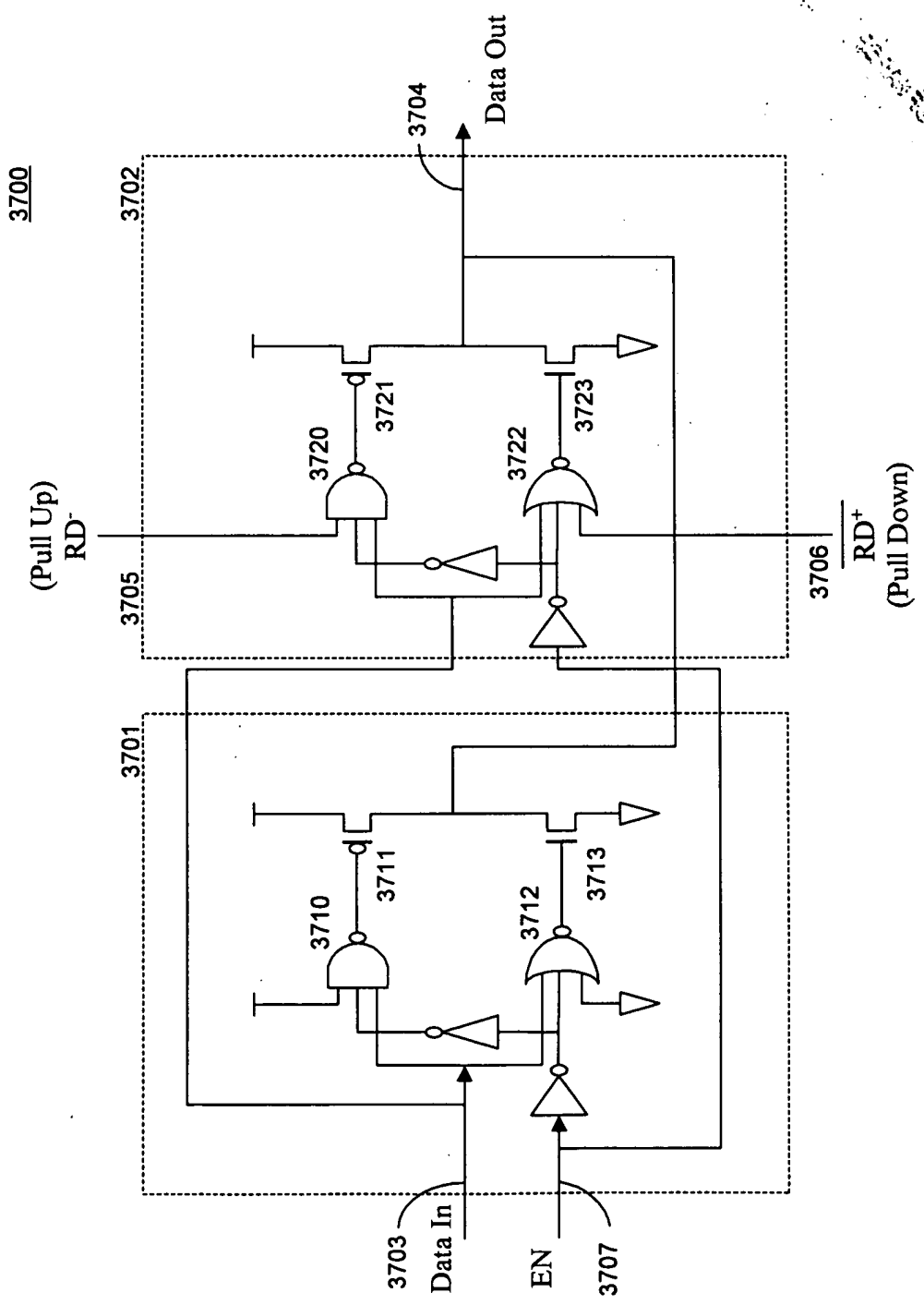

3700
3702
3704
Data Out
3705
(Pull Up)
RD-
3706
RD+
(Pull Down)
3701
3703
Data In
EN
3707
3710
3711
3712
3713
3720
3721
3722
3723

Fig. 37B

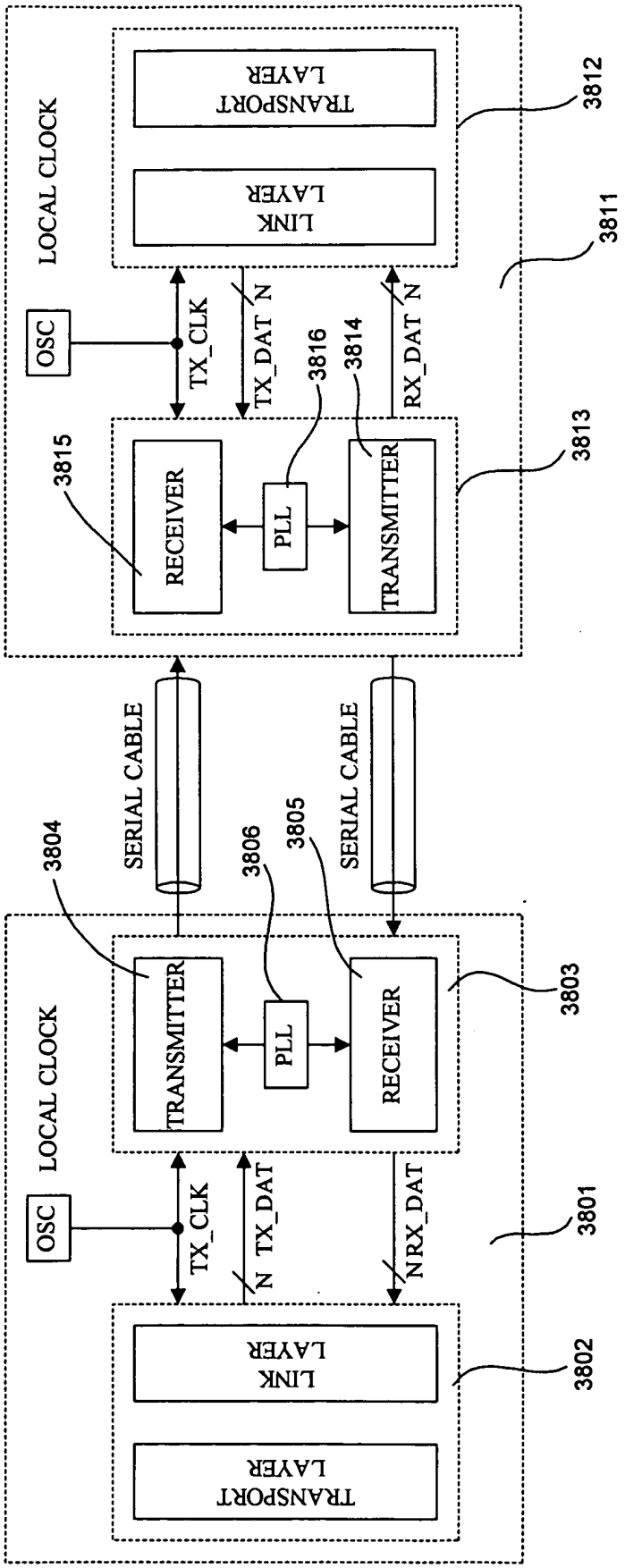

LOCAL CLOCK
TRANSPORT LAYER
LINK LAYER
3812
3811
OSC
TX_CLK
TX_DAT N
RX_DAT N
3816
3814
3815
RECEIVER
PLL
TRANSMITTER
3813
SERIAL CABLE
SERIAL CABLE
3804
3806
3805
LOCAL CLOCK
TRANSMITTER
PLL
RECEIVER
3803
OSC
TX_CLK
N TX_DAT
N RX_DAT
3801
LINK LAYER
TRANSPORT LAYER
3802

Fig. 38A

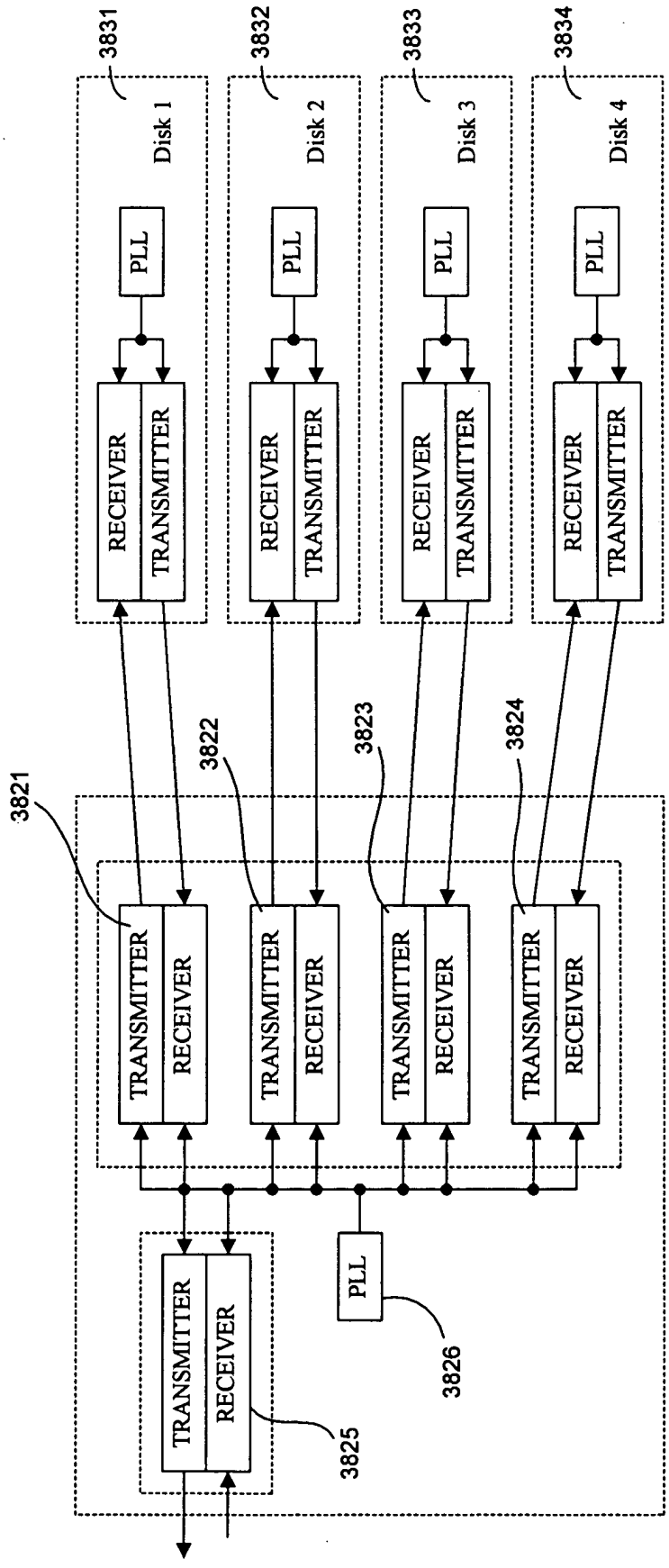

3831
Disk 1
PLL
RECEIVER
TRANSMITTER
3832
Disk 2
PLL
RECEIVER
TRANSMITTER
3833
Disk 3
PLL
RECEIVER
TRANSMITTER
3834
Disk 4
PLL
RECEIVER
TRANSMITTER
3821
3822
3823
3824
TRANSMITTER
RECEIVER
TRANSMITTER
RECEIVER
TRANSMITTER
RECEIVER
TRANSMITTER
RECEIVER
PLL
3826
TRANSMITTER
RECEIVER
3825

Fig. 38B

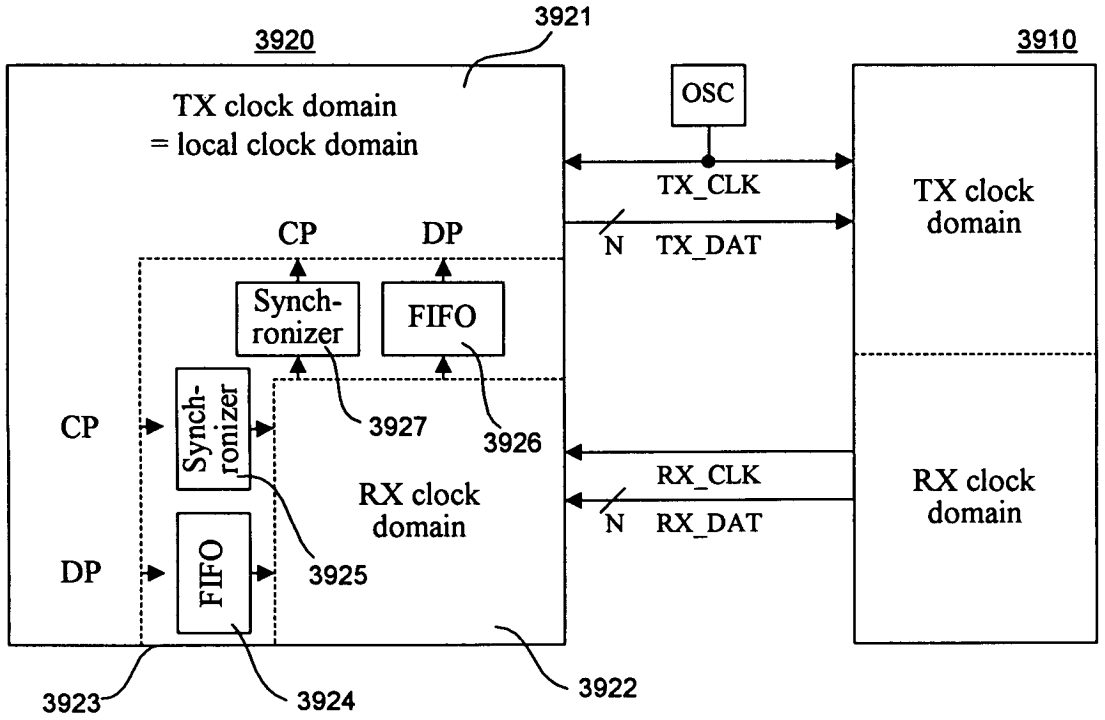


Fig. 39A

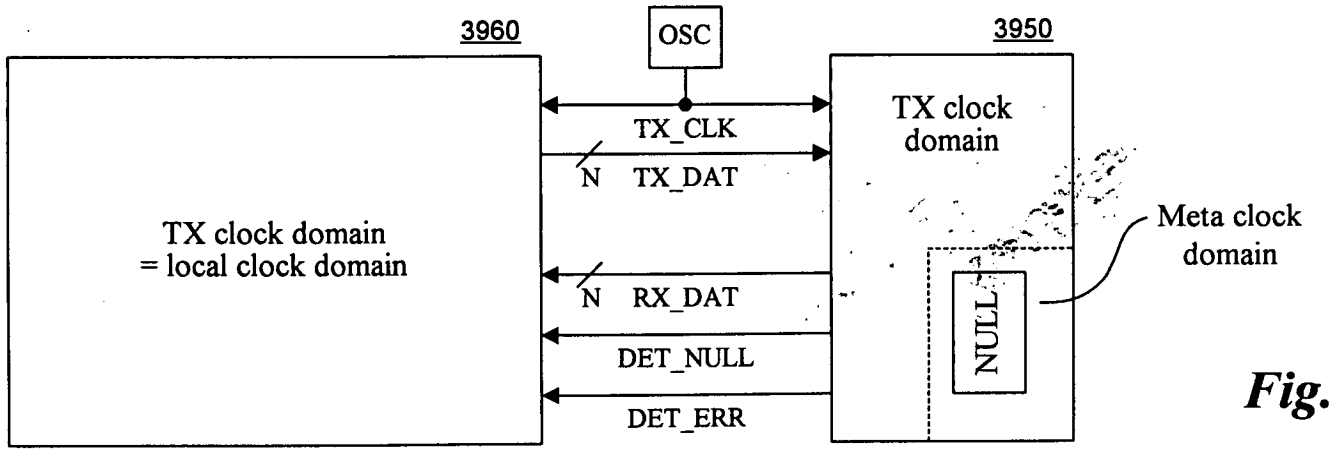


Fig. 39B

10053461 - 052202

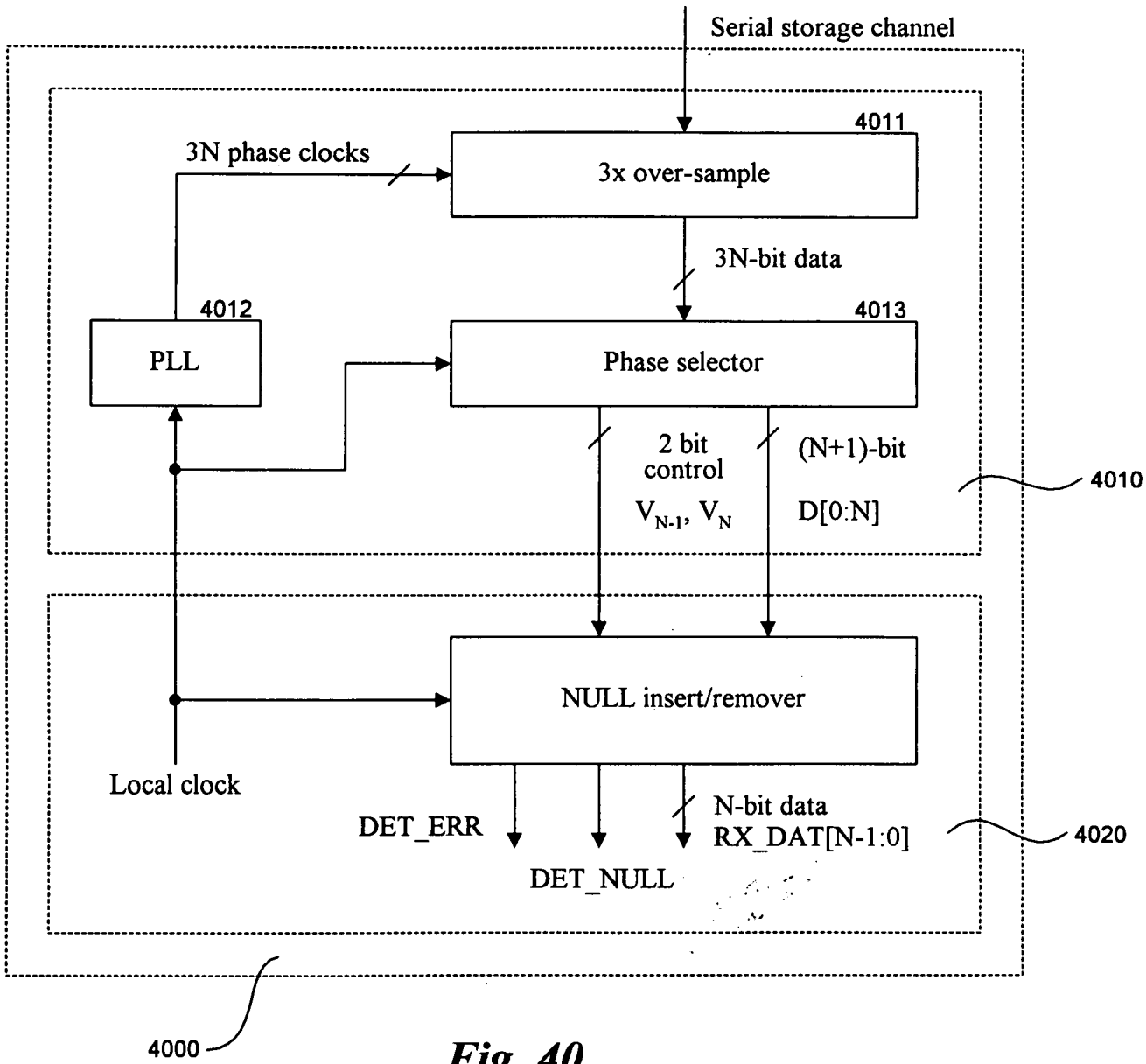


Fig. 40

20050461.050102

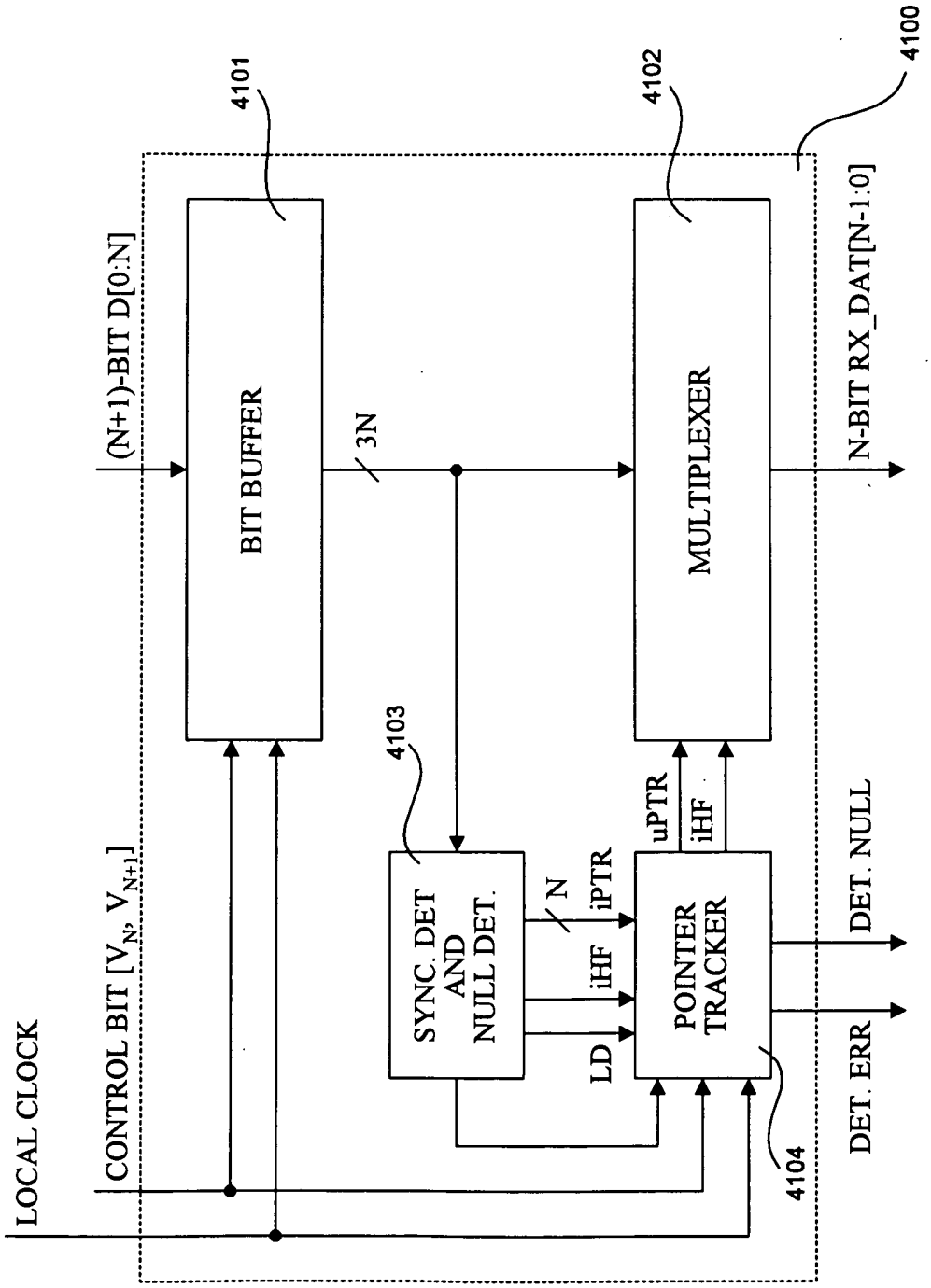

4100
4101
4102
4103
4104
BIT BUFFER
MULTIPLEXER
SYNC. DET AND NULL DET.
POINTER TRACKER
(N+1)-BIT D[0:N]
3N
N-BIT RX_DAT[N-1:0]
CONTROL BIT [V_N, V_{N+1}]
LOCAL CLOCK
N
iPTR
iHF
LD
uPTR
iHF
DET. NULL
DET. ERR

Fig. 41

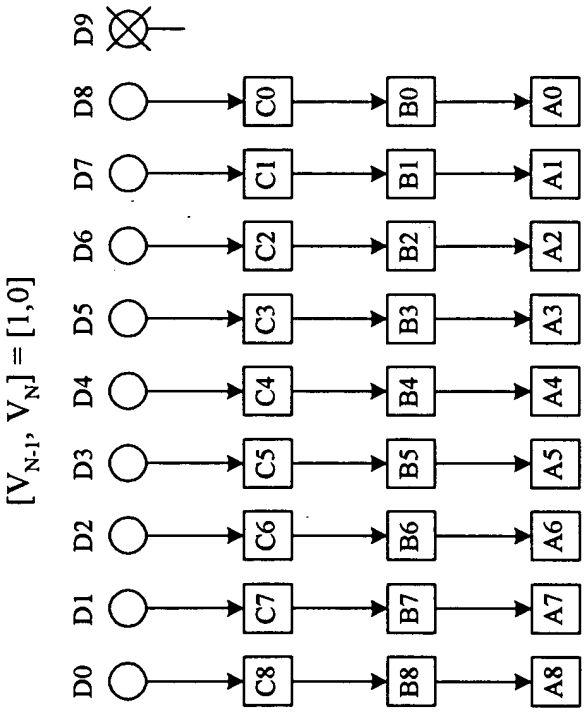

D9
D8
D7
D6
D5
D4
D3
D2
D1
D0
C0
C1
C2
C3
C4
C5
C6
C7
C8
B0
B1
B2
B3
B4
B5
B6
B7
B8
A0
A1
A2
A3
A4
A5
A6
A7
A8
[V_N-1, V_N] = [1,0]

Fig. 42A

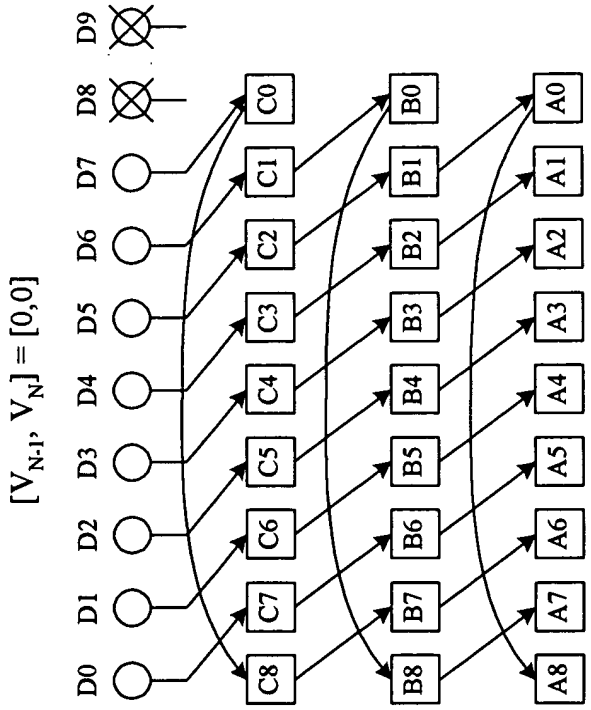

[V_{N-1}, V_N] = [0,0]
D0
D1
D2
D3
D4
D5
D6
D7
D8
D9
C0
C1
C2
C3
C4
C5
C6
C7
C8
B0
B1
B2
B3
B4
B5
B6
B7
B8
A0
A1
A2
A3
A4
A5
A6
A7
A8

Fig. 42B

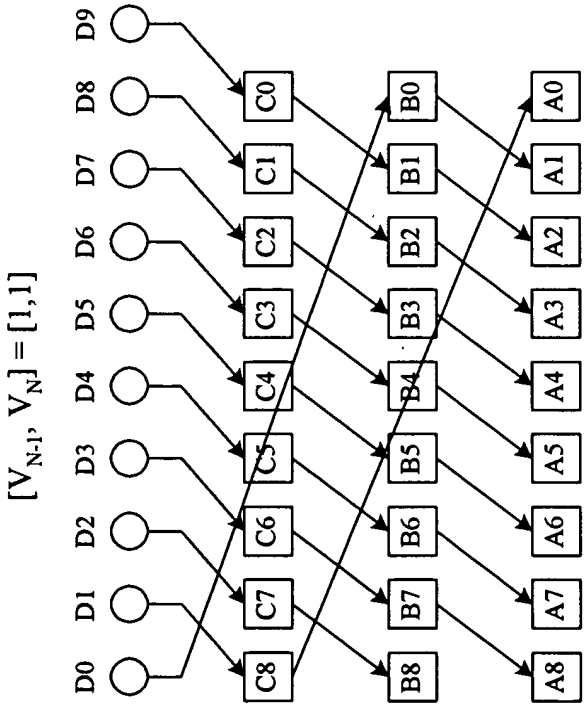

$[V_{N-1}, V_N] = [1,1]$
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9
C8 C7 C6 C5 C4 C3 C2 C1 C0
B8 B7 B6 B5 B4 B3 B2 B1 B0
A8 A7 A6 A5 A4 A3 A2 A1 A0

Fig. 42C

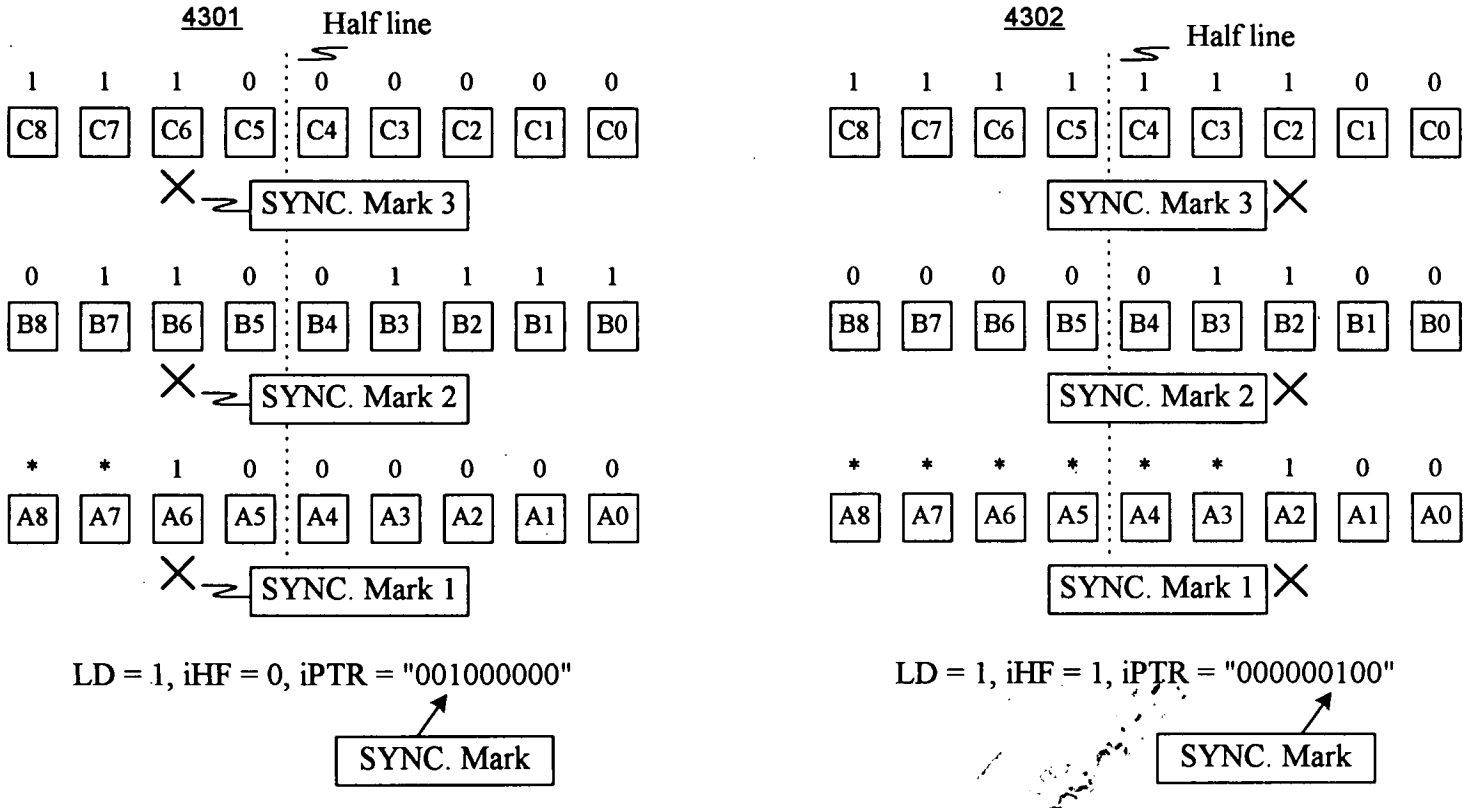
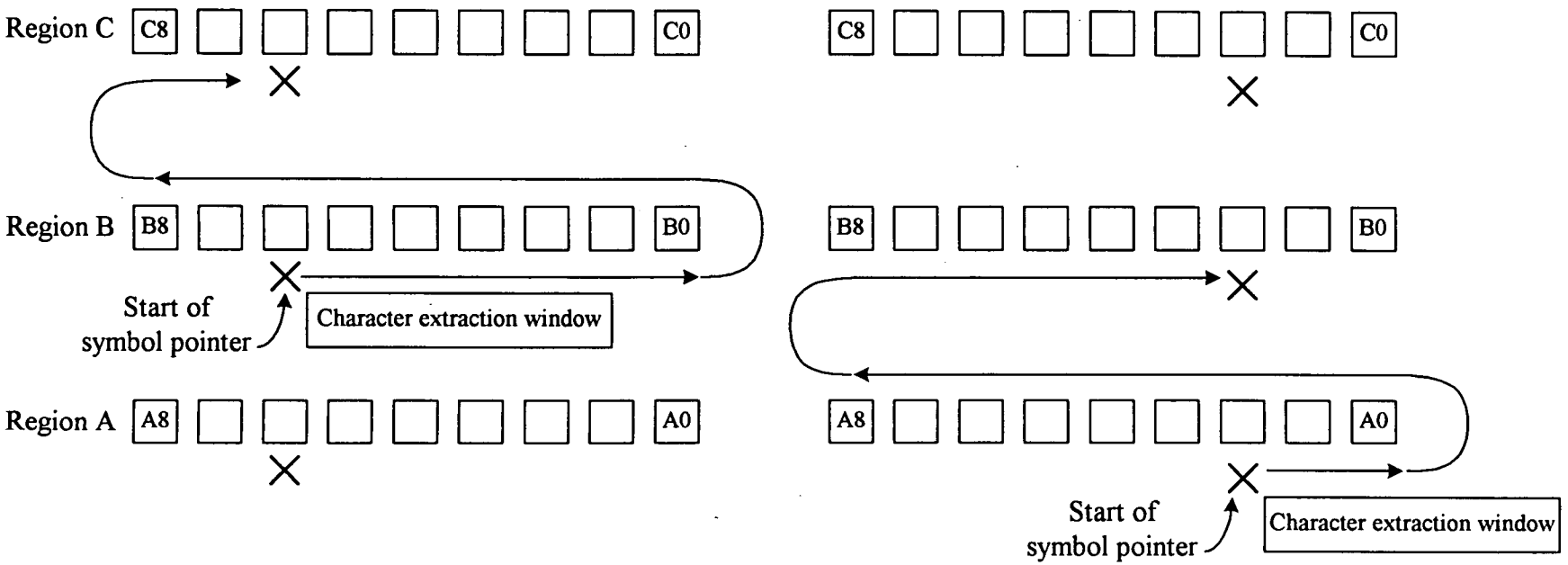


Fig. 43

1005346L-050202



LD = 1, iHF = 0, iPTR = "001000000"

LD = 1, iHF = 1, iPTR = "000000100"

Fig. 44

10053451 - 051102

10053461.051002

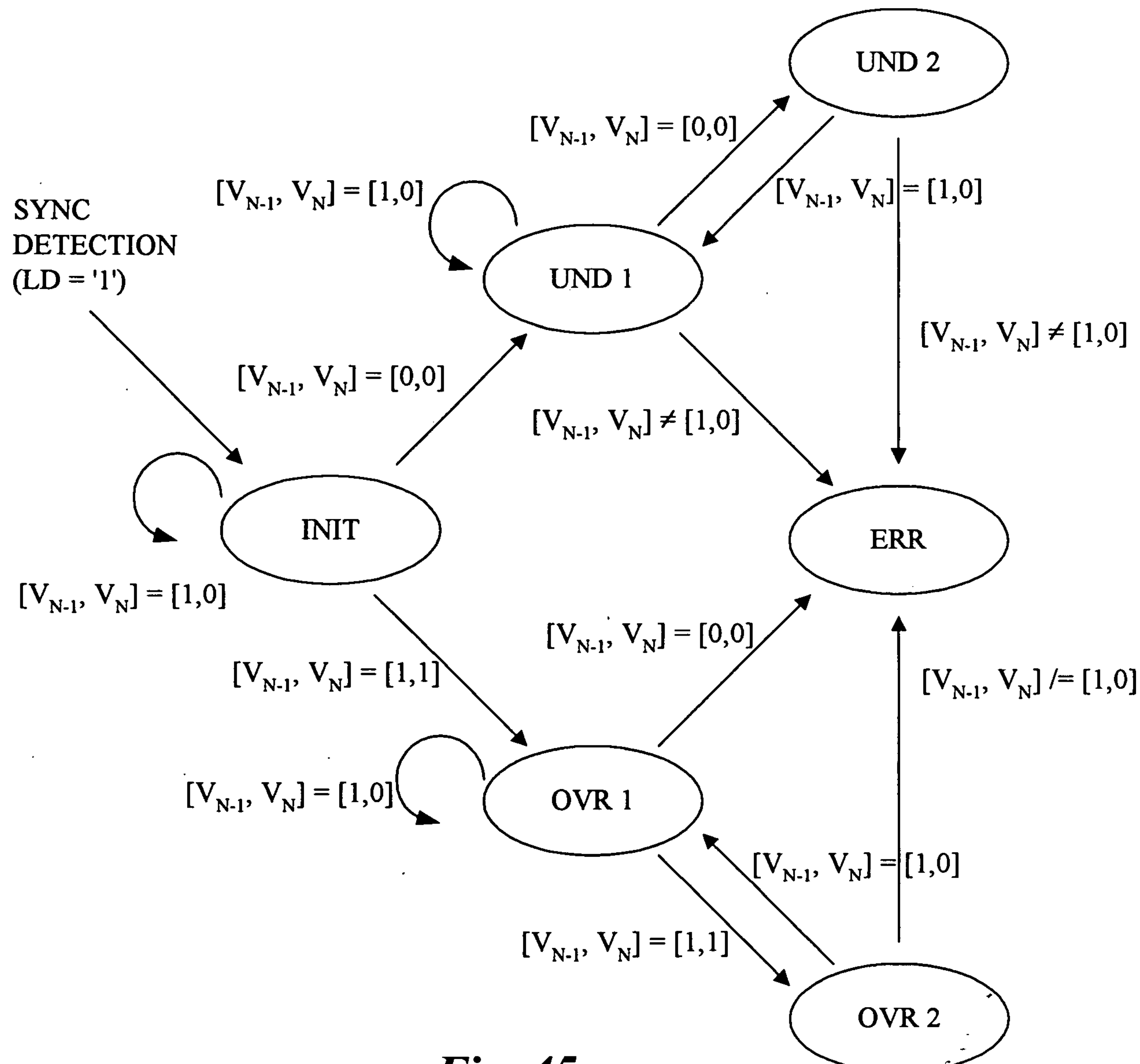


Fig. 45

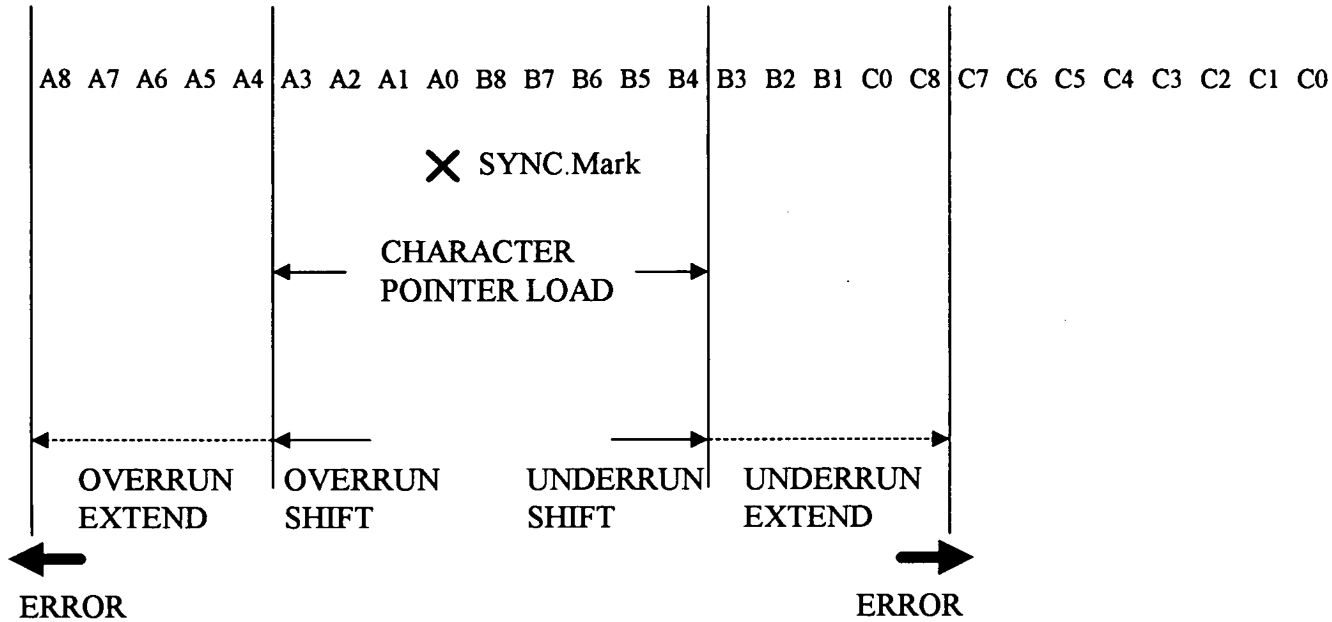


Fig. 46

10053461.051002

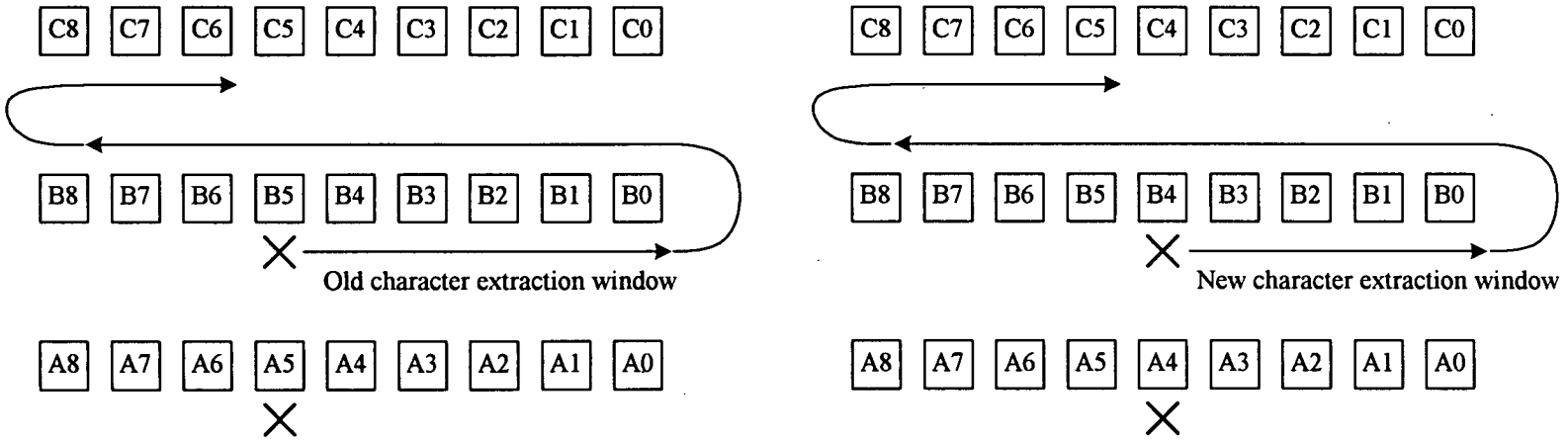


Fig. 47A

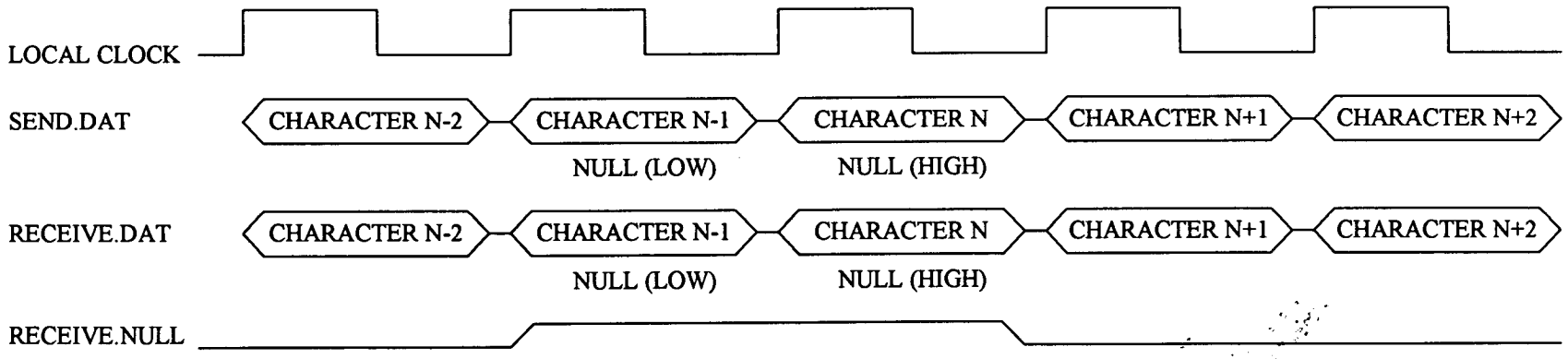


Fig. 47B

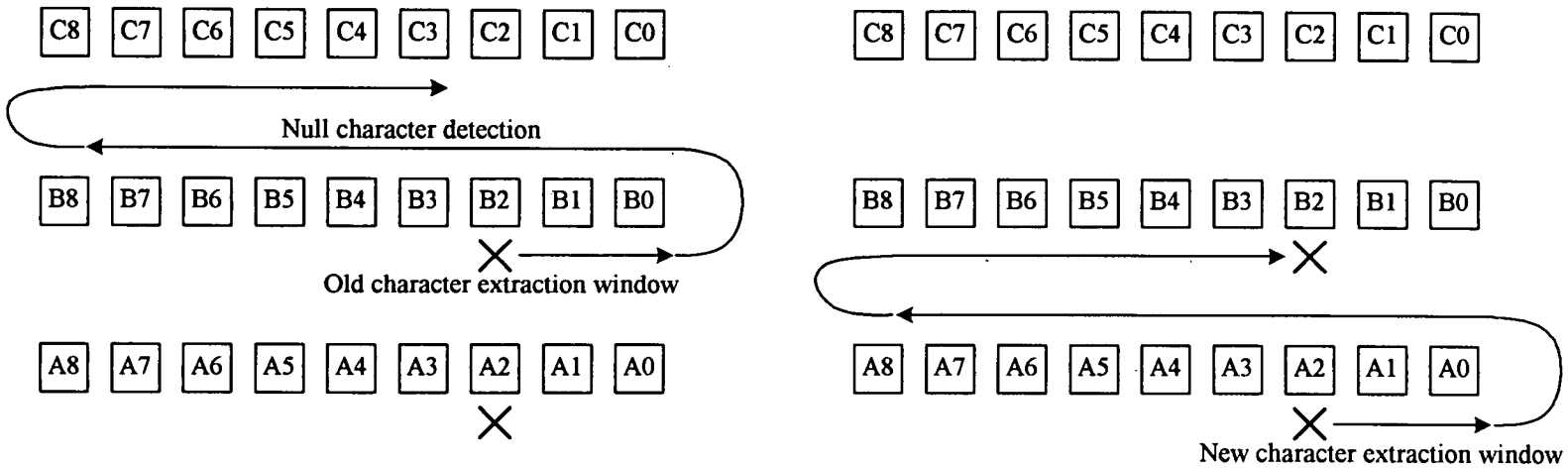


Fig. 48A

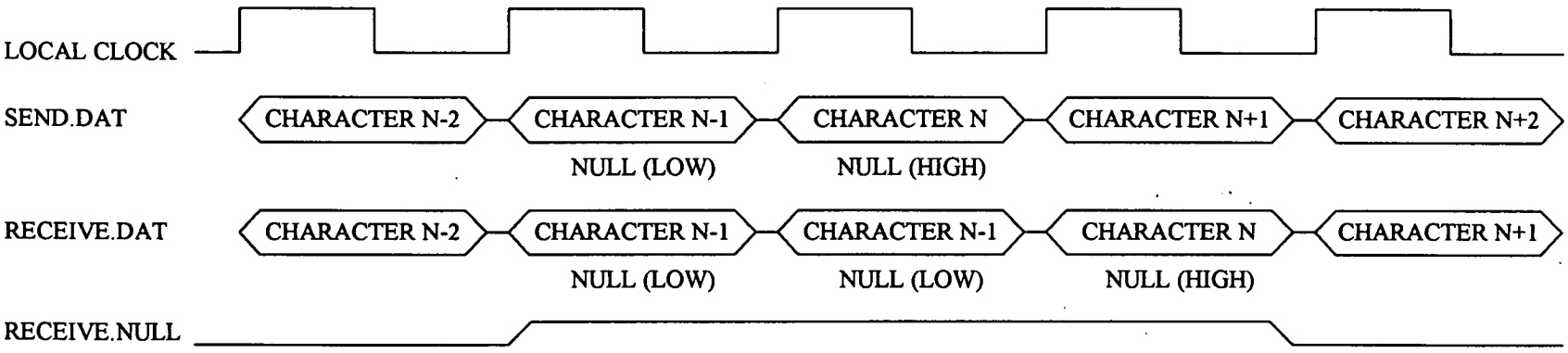


Fig. 48B

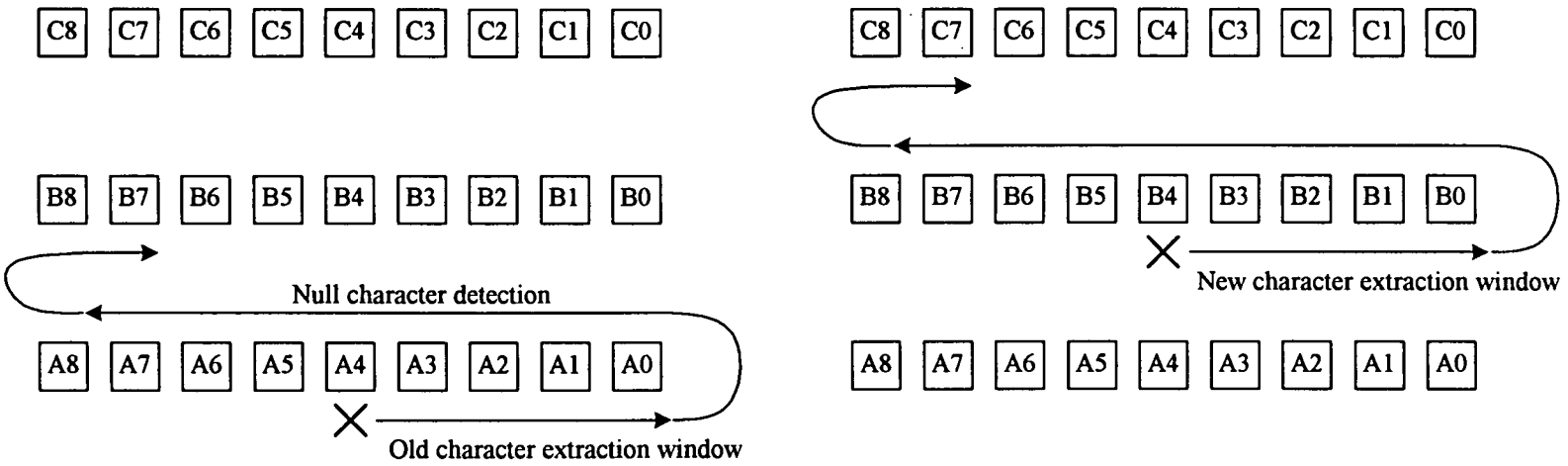


Fig. 49A

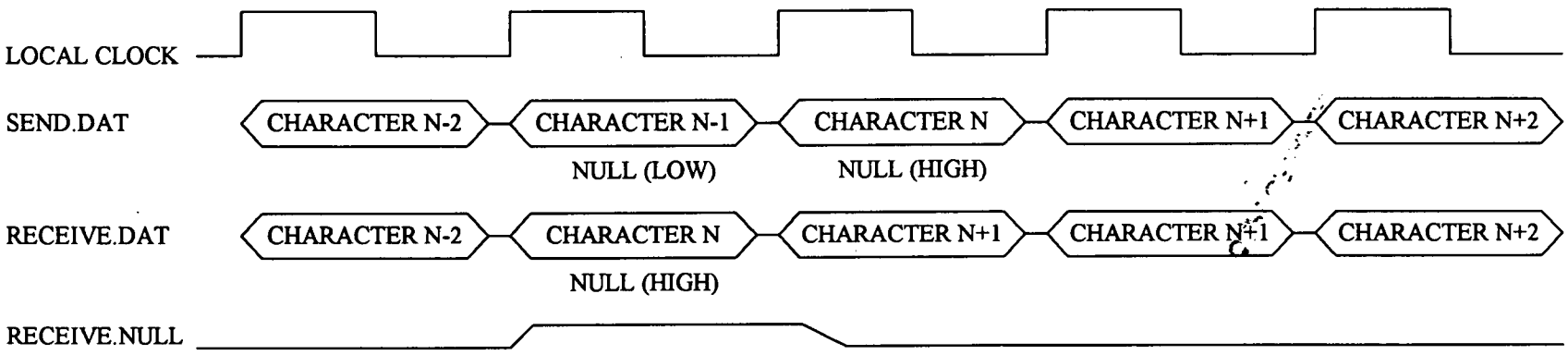


Fig. 49B